Leakage Resilient Leveled FHE on Multiple Bits Message

Zengpeng Li, *Student Member, IEEE*, Chunguang Ma, and Ding Wang

Abstract—Fully Homomorphic Encryption (FHE) allows computing over encrypted data without decrypting the corresponding ciphertexts, and it constitutes a promising cryptographic primitive to preserve data privacy in the big data computing environments. In general, FHE schemes can be constructed by using the standard Learning with Errors (LWE) assumption, and the current crux lies in how to achieve efficient multi-bit FHE encryption while being leakage-resistent against attackers who may capture the information of cryptographic secret keys via side channel attacks. Based on Berkoff-Liu's work at TCC'14, we aim to address this issue by giving a new structure of public key matrix with any number of LWE instances, thereby avoiding the use of a straightforward composition to achieve multi-bit FHE encryption under standard LWE. Particularly, our scheme attains provable security.

Index Terms—Big data, leakage resilient, leveled homomorphic encryption, learning with errors, multi-bit encryption

1 Introduction

Big data technologies collect vast amounts of data from heterogeneous sources to gain greater insight into patterns and trend not generally discernible when analyzing smaller data sets. Cloud computing and big data are conjoined, and their relationship can be well expressed by "Big data provides users the ability to use commodity computing to process distributed queries across multiple datasets and return resultant sets in a timely manner. Cloud computing provides the underlying engine through the distributed data-processing platforms like Hadoop" [1].

Although cloud computing brings many attractive opportunities, big data proved to be the "killer application" for the cloud computing that made it all work. However, without the right security and encryption solution in place, big data can mean serious problems. Namely, the main challenge arises naturally is that wherever there needs to be a trust boundary between data owners and computation-service providers, since once the data and operations are outsourced by the cloud server, the user lacks a valid mechanism for maintaining confidentiality and availability of the data [2], [3]. Following the trajectory of recent developments, a natural question that arises is:

How to overcome big data privacy and data insecurity in the cloud and prevent an attacker from modifying the highly-sensitive data and using those changes to gain unauthorized access to data?

Actually, the solution which provides mathematical guarantees of privacy in this setting, without the requirement to trust a third party's hardware, can only be provided by cryptography. To the best of our knowledge, fully homomorphic encryption (FHE) is one way to solve the above question, and it is an important component of a well-designed big data deployment. FHE allows the cloud server to perform operations on encrypted data without ever gaining any information about the processed data, that can protect data from unauthorized tampering and theft by malicious attackers.

FHE is a public key encryption scheme supporting algebraic operations on encrypted data. Specially, FHE is a secure homomorphic mapping from plaintext space to ciphertext space, allowing us to evaluate directly any function over encrypted data by only using public information, and such that the output is a ciphertext of the equivalent function over the corresponding plaintexts. Since the seminal work of Gentry [4] at STOC'09, many FHE schemes have been proposed [5], [6], [7], [8], [9], [10], [11] etc. Notably, the FHE scheme of Gentry, Sahai and Waters [6] (GSW) has proven to be useful breakthrough and the best current scheme.

1.1 Challenges and Motivations

A common feature of these traditional FHE schemes is that they only allow computation on encrypted single-bit, and the efficiency is unsatisfactory. Most of FHE constructions focus on single bit encryption, though it is pointed out they can be amortized in the concatenation way (or a straightforward composition) to achieve multi-bit computation. Unfortunately, this straightforward construction for multi-bit FHE will not lead to the best performance. To alleviate the efficiency problem, Hiromasa et al. [9] constructed the multi-bit FHE at PKC'15 by packing message. Essentially, their scheme only allows homomorphic

- *Z. Li and C. Ma are with the College of Computer Sciences and Technology, Harbin Engineering University, Harbin 150001, P.R. China, and the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100864, China. E-mail: {lizengpeng, machungaung}@hrbeu.edu.cn.*
- *D. Wang is with the School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, P.R. China. E-mail: wangdingg@pku.edu.cn.*

Manuscript received 1 Dec. 2016; revised 31 Mar. 2017; accepted 14 May 2017. Date of publication 13 July 2017; date of current version 27 Sept. 2021. (Corresponding author: Ding Wang.)
Recommended for acceptance by S. Yu, P. Mueller, J. Pei, and K. Liu.
Digital Object Identifier no. 10.1109/TBDATA.2017.2726554

decryption on encrypted matrix bit-by-bit, which is very inefficient. On the other hand, multi-key FHE [10], [12], [13], [14], [15] is another way to improve the efficiency of homomorphic evaluation, but it is outside the scope of this paper.

In reality, many scenarios require computations on multi-bit data at one-time. Very recently, the line of the research has been improved—in [9], [16], the authors proposed interesting new ideas to construct FHE for multi-bit messages, and further improved the efficiency of FHE. However, these schemes are constructed via the straightforward construction and will not get the best performance. More concretely, **1).** the scheme of Brakerski et al. [16] was constructed on top of Brakerski [5] scheme which is the representation scheme of the second generation FHE, but the Brakerski [5] scheme needs the evaluation key to achieve homomorphic evaluation which leads to the increase of the computational overhead; **2).** The scheme of Hiromasa-Abe-Okamoto [9] (HAO) was constructed by using GSW [6] scheme as building block which is the representation scheme of the third generation FHE, but the HAO scheme cannot achieve one-time decryption and only decrypts the ciphertext in bit-by-bit.

This naturally raises an important question: *Is it possible to design an efficient method to construct* LWE*-based multi-bit GSW-FHE encryption rather than via straightforward concatenation?*

We will formally explore this important question in this paper and believe that, the multi-bit public key encryption based on the public key with many LWE instances might offer great advantages over other approaches.

What's more, it is realistic to assume that attackers can capture the information of cryptographic secret key via side channel attacks, and this issue has prompted cryptographers to construct "leakage resilient" schemes that remain secure under such attacks. The term "leakage resilient" is used to unite the theoretical cryptography with the practical truth of side channel attacks, and Micali-Reyzin managed to devise a theoretical model of cryptography that would cover all such attacks [17]. In particular, their goal was that if a scheme could be proved to be secure under their model, then it would also be secure against side channel attacks. In other words, it is meant to capture the security of cryptographic algorithm when an adversary uses non-standard methods to learn about the secret key [8]. Akavia et al. [18] further considered side-channel memory attacks against the LWE assumption. Techniques along this line have recently been further improved [18], [19], [20]. To the best of our knowledge, among all aforementioned schemes, none can constitute a leakage-resilient multi-bit FHE scheme. Such issues prompt us to ask the question:

How to construct a leveled FHE *scheme with multi-bit messages to be secure against side-channel attacks?*

1.2 Our Contributions

In this work, we first construct a public key matrix with multiple LWE instances rather than one LWE instance in the standard public key matrix like [5], [6], [14]. Then, utilizing the new public key, we construct a multi-bit GSW-style FHE scheme (hereafter MGSW). Furthermore, in the multi-bit setting, we construct a leveled, leakage-resilient FHE. More concretely,

- First, we propose a new structure of public key matrix with multiple LWE instances as follows:

  $$\mathbf{A}' = [\mathbf{b}_1, \ldots, \mathbf{b}_t \mid \mathbf{A}] \in \mathbb{Z}_q^{m\times(n+t)},$$

  where $\mathbf{b}_i = \mathbf{A}\cdot\mathbf{t}_i + \mathbf{e}_i \pmod q$ for $i \in [t]$ are LWE instances. This technique is significantly different from the existing multi-bit PKE schemes (e.g., [21], [22]) and multi-bit FHE schemes (e.g., [9], [16]) that were designed via the method of packed Regev's encryption [21].
- Second, we construct a leakage-resilient leveled FHE scheme (hereafter LRMGSW) via the methodology of Berkoff-Liu [8] (BL) under the "bounded adaptive leakage resilient" model. For better comparison, we further construct a leakage-resilient on HAO [9] scheme (we abbreviate it to LRHAO) via the methodology of Berkoff-Liu [8].

1.3 Paper Organization

The remainder of this paper is organized as follows. In Section 2 we formally present the notations that will be used throughout the paper. In Section 3 we describe our multi-bit FHE scheme (MGSW) via LWE assumption. In Section 4 we show that a variant of leakage resilient FHE scheme on long message (LRMGSW) via LWE assumption and we construct a leakage resilient Hiromasa et al. (PKC'15) scheme in Section 5. Finally, in Section 6 we give a conclusion.

2 Preliminaries

In this section we provide required preliminaries. We note that, the definitions and lemmas are taken from previous work.

2.1 Notation

For $n \in \mathbb{N}$, we let $[n]$ denote the set $\{1, \ldots, n\}$. For a real number $x \in \mathbb{R}$, we let $\lfloor x \rfloor$ denote the largest integer not greater than x, and $\lfloor x \rceil := \lfloor x + \frac{1}{2} \rfloor$ denote the integer closest to x, with ties broken upward. We use bold lower-case letters, e.g., $\mathbf{x}$, to denote vectors, and bold upper-case letters, e.g., $\mathbf{A}$, to denote matrices. Moreover, we use $\mathbf{A}_{i,j}$ to denote the element at ith row and jth column of $\mathbf{A}_{i,j}$. We use ":=" to denote deterministic assignment. Notably, we use the definition of computational indistinguishability and statistical indistinguishability, denoted by $\approx_c$ and $\approx_s$, separately. In addition, we also denote $\|\mathbf{v}\|_\infty = max\{|v_1|, \ldots, |v_n|\}$ and $\|\mathbf{R}\| = \max_i \|\mathbf{r}_i\|$. For notational convenience, we let $\|\mathbf{v}\|$ denote its l_2 norm.

We use the following variant of the Leftover Hash Lemma (LHL) [23].

Lemma 2.1 (Matrix-vector LHL [24] Lemma 2.1). *Let $\lambda \in \mathbb{Z}$, n, $q \in \mathbb{N}$, $m \geq n\log q + 2\lambda$, $\mathbf{r} \xleftarrow{R} \{0,1\}^m$ and $\mathbf{y} \xleftarrow{R} \mathbb{Z}_q^n$. Sample uniformly random matrix $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{m\times n}$, then the statistical distance between the distributions $(\mathbf{A}, \mathbf{A}^T\mathbf{r})$ and $(\mathbf{A}, \mathbf{y})$ is as follows:*

$$\Delta\big((\mathbf{A}, \mathbf{A}^T\cdot\mathbf{r}), (\mathbf{A}, \mathbf{y})\big) \leq 2^{-\lambda}. \quad (1)$$

2.2 Discrete Gaussians

In our constructions we need to analyze the behavior of error elements sampled from Gaussian distributions.

Definition 2.2 ([5] Def. 2.1). *A distribution ensemble $\chi = \chi(\lambda)$ over the integers is called B-bounded (denoted $|\chi| \leq B$) if there exists*

$$\Pr_{x \xleftarrow{\$} \chi}[|x| \geq B] \leq 2^{-\tilde{\Omega}(n)}.$$

For the analysis of our scheme, we require some bounds on the norms of vectors sampled from Gaussian distributions.

Lemma 2.3 ([25] Lemma 4.4).

1) *For $\forall\, k > 0$, $Pr[|e| > k \cdot \sigma, e \leftarrow D_\sigma^1] \leq 2 \cdot \exp(-\frac{k^2}{2})$;*
2) *For $\forall\, k > 0$, $Pr[\|\mathbf{e}\| > k \cdot \sigma \cdot \sqrt{m},\ \mathbf{e} \leftarrow D_\sigma^m] \leq k^m \cdot \exp(\frac{m}{2} \cdot (1 - k^2))$.*

Remark 2.4. Throughout the paper, we suppose $\sigma \geq 2\sqrt{n}$. Therefore, if $\mathbf{e} \leftarrow D_\sigma^m$ then we have, on average, that $\|\mathbf{e}\| \approx \sqrt{m} \cdot \sigma$. Lemma 2.3 (2) implies that $\|\mathbf{e}\| \leq 2\sigma\sqrt{m}$ with overwhelming probability. Hence, in this paper, we set $|e| \leq B$ and $\|\mathbf{e}\| \leq 2\sqrt{m}B$.

2.3 Min-Entropy Notions

Definition 2.5 ([26]). *For $0 < \mu < \frac{1}{2}$, we denote the binary entropy function as $H(\mu) \stackrel{def}{=} \mu \log 1/\mu + (1-\mu)\log 1/(1-\mu)$; the Shannon entropy of a random variable X is as follows:*

$$H_1(X) \stackrel{def}{=} \sum_{x \in Sup(X)} Pr[X = x] \log \frac{1}{Pr[X = x]};$$

and the min-entropy of a random variable X is as follows:

$$H_\infty(X) \stackrel{def}{=} \min_{x \in Sup(X)} \log \frac{1}{Pr[X = x]}.$$

We also denote a distribution $\mathcal{X}$ which has min entropy $H_\infty(\mathcal{X}) \geq \lambda$ for $\forall\, x \in \mathcal{X}$, there exists $Pr[\mathcal{X} = x] \leq 2^{-\lambda}$.

Definition 2.6 ([8] Def 15 from [27]). *For two random variables X and Y, the average min-entropy of X conditioned on Y, denoted*

$$\begin{aligned}\tilde{H}_\infty(X \,|\, Y) &:= -\log \mathbf{E}_{y \leftarrow Y}\Big[[\max_x Pr[X = x \,|\, Y = y]]\Big] \\ &= -\log\Big[\mathbf{E}_{y \leftarrow Y}[2^{-H_\infty(X \,|\, Y = y)}]\Big].\end{aligned}$$

Definition 2.7 ([8] Def 16 from [27]). *For two random variables X and Y, the ε-smooth average min-entropy of X conditioned on Y, for $\Delta\big((X,Y),(X',Y')\big) < \varepsilon$ denoted*

$$\tilde{H}_\infty(X \,|\, Y) = \max_{(X',Y'):\Delta < \varepsilon} \tilde{H}_\infty(X' \,|\, Y').$$

Most notably, for any random variables X, given the distributions $\mathcal{D}_Y \approx_s \mathcal{D}_Z$ with $Y \leftarrow \mathcal{D}_Y$, and $Z \leftarrow \mathcal{D}_Z$, there exist some ε such that $\Delta(Y,Z) < \varepsilon = negl(\eta)$, and $\tilde{H}_\infty(X \,|\, Y) \geq \tilde{H}_\infty(X \,|\, Z)$.

2.4 Learning with Errors (LWE)

The learning with errors problem is the main computational assumption underlying the GSW cryptosystem and our variant of it.

Definition 2.8 (LWE Distribution). *For the security parameter λ, let $n = n(\lambda)$ and $m = m(\lambda)$ be integers, let $\chi = \chi(\lambda)$ be error distribution over $\mathbb{Z}$ bounded by $B = B(\lambda)$, and let $q = q(\lambda) \geq 2$ be an integer modulus for any polynomial $p = p(\lambda)$ such that $q \geq 2^p \cdot B$. Then, sample a vector $\mathbf{s} \in \mathbb{Z}_q^{n \times 1}$ called the secret, the LWE distribution $\mathcal{A}_{\mathbf{s},\chi}$ over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ is sampled by choosing $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ uniformly at random, choosing $\mathbf{e} \leftarrow \chi^{m \times 1}$, and outputting $\big(\mathbf{A}, \mathbf{b} = \mathbf{A} \cdot \mathbf{s} + \mathbf{e} \,(\mathrm{mod}\ q)\big)$.*

There are two main versions of the LWE assumption: search version and decision version. We define the decision version as follows,

Definition 2.9 (Decision-LWE$_{n,q,\chi,m}$). *Assume given an independent sample $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^{m \times 1}$, where the sample is distributed according to either: (1) $\mathcal{A}_{\mathbf{s},\chi}$ for a uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$ (i.e., $\{(\mathbf{A}, \mathbf{b}) : \mathbf{A} \leftarrow \mathbb{Z}_q^{m \times n}, \mathbf{s} \leftarrow \mathbb{Z}_q^{n \times 1}, \mathbf{e} \leftarrow \chi^{m \times 1}, \mathbf{b} = \mathbf{A} \cdot \mathbf{s} + \mathbf{e} \,(\mathrm{mod}\ q)\}$), or (2) the uniform distribution (i.e., $\{(\mathbf{A}, \mathbf{b}) : \mathbf{A} \leftarrow \mathbb{Z}_q^{m \times n}, \mathbf{b} \leftarrow \mathbb{Z}_q^{m \times 1}\}$). Then, the above two distributions are computationally indistinguishable.*

Remark 2.10. Regev and others [10], [28], [29], [30] show the reductions between the LWE assumption and approximating the shortest vector problem in lattices (for appropriate parameters). We omit the corollary of these schemes' results and refer to find further details from [10], [28], [29], [30].

2.5 Leveled Fully Homomorphic Encryption

In a public-key encryption, the encrypter holds a public key and encrypts a message such that the holder of the corresponding secret key is able to reconstruct the original plaintext message. Meanwhile, the ciphertext can be evaluated homomorphically.

Definition 2.11 ([31]). *Fix a function $L = L(\lambda)$. An L-FHE scheme for a class of circuits $\{\mathcal{C}_\lambda\}_{\lambda \in N}$ consists of four probabilistic polynomial-time (PPT) algorithms {KeyGen, Enc, Dec, Eval} such that:*

- *key generation algorithm KeyGen is a randomized algorithm that takes the security parameter 1^λ as input and outputs a public key pk and a secret key sk;*
- *encryption algorithm Enc is a randomized algorithm that takes a public key pk and a message $m \in \{0,1\}^*$ as input, and outputs a ciphertext c;*
- *decryption algorithm Dec is a deterministic algorithm that takes the secret key sk and a ciphertext c as input, and outputs a message $m \in \{0,1\}^*$;*
- *homomorphic evaluation algorithm Eval takes as input a public key pk, a circuit $C \in \mathcal{C}_\lambda$, and a list of ciphertexts $c_1, \ldots, c_\ell$, where ℓ is polynomial over λ, and it outputs a ciphertext c^*.*

The correctness properties are required as follows:

- for any λ, $m \in \{0,1\}^*$, and $(\mathsf{pk}, \mathsf{sk})$ output by $\mathsf{KeyGen}(1^\lambda)$, we have that
$$m = \mathsf{Dec}\Big(\mathsf{sk}, \big(\mathsf{Enc}(\mathsf{pk}, m)\big)\Big);$$

- for any λ, any $m_1, \ldots, m_l \in \{0,1\}^*$, and $C \in \mathcal{C}_\lambda$, we have that

$$\mathcal{C}(m_1, \ldots, m_\ell) = \mathsf{Dec}\Big(\mathsf{sk}, \big(\mathsf{Eval}\big(\mathsf{pk}, (C, \mathsf{Enc}(\mathsf{pk}, m_1), \ldots, \mathsf{Enc}(\mathsf{pk}, m_\ell))\big)\big)\Big).$$

Definition 2.12 ([31]). *A* FHE *scheme is indistinguishable* chosen-plaintext attacks *(*IND-CPA*)-secure if we have that for any* PPT *adversary $\mathcal{A}$ the following is negligible in λ*

$$|\Pr[\mathcal{A}(\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, m_0)) = 1] - \Pr[\mathcal{A}(\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, m_1)) = 1]|,$$

where $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$ and m_0, m_1 are arbitrarily chosen from the plaintext space by the adversary.

The security definition for multi-bit GSW (MGSW) is the same as that for single-bit GSW. Since in the public key setting, security for encryption of a single message implies security for encryption of multiple message. More details see Chapter 11 of [32].

Definition 2.13 (Compactness, [31] Def. 3). *An L-FHE scheme for a class of circuits $\{\mathbb{C}_k\}_{k\in\mathbb{N}}$ is* compact *if there exists a polynomial $\alpha = \alpha(\lambda)$ such that ciphertexts output by* Eval *have length at most α. (For this to be non-trivial it should be the case that, for all λ, we have $\alpha(\lambda) \le |C|$ for some $C \in \{\mathbb{C}\}_\lambda$).*

2.6 Basic Tools

Below we review some basic tools proposed by Brakerski et al. [24] and Gentry et al. [6]. Fix $q, m \in \mathbb{N}$. Let $l = \lfloor \log(q) \rfloor + 1$, so that $2^{l-1} \le q < 2^l$, and $N = m \cdot l$.

Definition 2.14 ([13], [33]). *The algorithm* $\mathsf{BitDecomp}$ *takes as input a vector $\mathbf{v} \in \mathbb{Z}_q^m$ and outputs an N dimensional vector $(v_{1,0}, \ldots, v_{1,l-1}, \ldots, v_{m,0}, \ldots, v_{m,l-1})^T \in \{0,1\}^N$ where $v_{i,j}$ is the jth bit in v_i's binary representation (ordered from least significant to most significant). In other words,*

$$v_i = \sum_{j=0}^{l-1} 2^j v_{i,j}.$$

Definition 2.15([13], [33]). *The algorithm* $\mathsf{BitDecomp}^{-1}$ *takes as input a vector*

$$\mathbf{v} = (v_{1,0}, \ldots, v_{1,l-1}, \ldots, v_{m,0}, \ldots, v_{m,l-1})^T \in \mathbb{Z}_q^N,$$

and outputs $(\sum_{j=0}^{l-1} 2^j \cdot v_{1,j}, \ldots, \sum_{j=0}^{l-1} 2^j \cdot v_{m,j})^T \in \mathbb{Z}_q^m$. Note that the input vectors $\mathbf{v}$ need not be binary, the algorithm is well-defined for any input vector in $\mathbb{Z}^N$.

Definition 2.16 ([13], [33]). *The algorithm* $\mathsf{Flatten}$ *takes as input a vector $\mathbf{v} \in \mathbb{Z}_q^N$ and outputs an N dimensional binary vector (i.e., an element of $\{0,1\}^N$). It is defined by*

$$\mathsf{Flatten}(\mathbf{v}) = \mathsf{BitDecomp}(\mathsf{BitDecomp}^{-1}(\mathbf{v})).$$

Definition 2.17 ([13], [33]). *The algorithm* $\mathsf{PowerOf2}$ *takes an m dimensional vector $\mathbf{v} \in \mathbb{Z}_q^m$ and outputs an N dimensional vector in $\mathbb{Z}_q^N$ as follows:*

$$\big(v_1, 2v_1, \ldots, 2^{l-1}v_1, \ldots, v_m, 2v_m, \ldots, 2^{l-1}v_m\big)^T.$$

Lemma 2.18 ([34] and [10] Lemma 2.1). *For any $N \ge m\lceil \log q \rceil$ there exists a fixed efficiently computable matrix $\mathbf{G} \in \mathbb{Z}_q^{m\times N}$ and an efficiently computable deterministic "short preimage" function $\mathbf{G}^{-1}(\cdot)$ satisfying the following. On input a matrix $\mathbf{M} \in \mathbb{Z}_q^{m\times m'}$ for any m'. The inverse function $\mathbf{G}^{-1}(\mathbf{M})$ outputs a matrix $\mathbf{G}^{-1}(\mathbf{M}) \in \{0,1\}^{N\times m'}$ such that $\mathbf{G}\mathbf{G}^{-1}(\mathbf{M}) = \mathbf{M}$.*

Remark 2.19. Actually, the above definitions and results can therefore be expressed in the language of $\mathbf{G}$ and $\mathbf{G}^{-1}$ as follows. The gadget matrix $\mathbf{G}$ from Micciancio and Peikert [34] can be expressed by $\mathbf{G} = \mathbf{I}_m \otimes \mathbf{g} \in \mathbb{Z}_q^{m\times N}$ where $\mathbf{g} = (1, 2, 4, \ldots, 2^{l-1})^T$. For $\mathbf{v} \in \mathbb{Z}_q^m$ we have $\mathsf{PowerOf2}(\mathbf{v}) = \mathbf{v}^T\mathbf{G}$. For $\mathbf{v} \in \mathbb{Z}_q^N$ we have $\mathsf{BitDecomp}^{-1}(\mathbf{v}) = \mathbf{G}\mathbf{v}$. For $\mathbf{a} \in \mathbb{Z}_q^m$ the algorithm $\mathsf{BitDecomp}(\mathbf{a})$ can be renamed as $\mathbf{G}^{-1}(\mathbf{a})$.

3 FHE on Multi-Bit Scheme

Before describing our construction, we first review the Gentry-Sahai-Waters homomorphic encryption scheme, then we sketch the correctness and security due to Gentry et al. [6].

3.1 Gentry-Sahai-Waters Scheme

Let λ be a security parameter and let L be the number of levels for the somewhat homomorphic scheme. We describe the algorithms that form the GSW scheme [6]. The algorithms are originally defined in terms of the functions $\mathsf{BitDecomp}, \mathsf{BitDecomp}^{-1}$ and $\mathsf{Flatten}$, but we tend to follow the formulation in [7], [10] and so use the matrix $\mathbf{G}$.

- $\mathsf{GSW.Setup}(1^k, 1^L)$:
  1) Choose a modulus q of $\kappa = \kappa(\lambda, L)$ bits, parameter $n = n(\lambda, L) \in \mathbb{N}$, and error distribution $\chi = \chi(\lambda, L)$ on $\mathbb{Z}$ so that the (q, n, χ)-LWE problem achieves at least 2^k security against known attacks.
     Choose a parameter $m = m(\lambda, L) = O(n\log(q))$;
  2) Output: $\mathsf{params} = (n, q, \chi, m)$.
     We denote $l = \lfloor \log(q) \rfloor + 1$ and $N = (n+1) \cdot l$.
- $\mathsf{GSW.KeyGen}(\mathsf{params})$:
  1) Sample $\mathbf{t} = (t_1, \ldots, t_n)^T \leftarrow \mathbb{Z}_q^n$ and compute
     $$\mathbf{s} \leftarrow (1, -\mathbf{t}^T)^T = (1, -t_1, \ldots, -t_n)^T \in \mathbb{Z}_q^{(n+1)\times 1};$$
  2) Generate a matrix $\mathbf{B} \leftarrow \mathbb{Z}_q^{m\times n}$ and a vector $\mathbf{e} \leftarrow \chi^m$;
  3) Compute $\mathbf{b} = \mathbf{B}\mathbf{t} + \mathbf{e} \in \mathbb{Z}_q^m$ and construct the matrix $\mathbf{A} = (\mathbf{b} \,|\, \mathbf{B}) \in \mathbb{Z}_q^{m\times(n+1)}$ as the vector $\mathbf{b}$ followed by the n columns of $\mathbf{B}$.
     Apparently, we observe that
     $$\mathbf{A}\mathbf{s} = (\mathbf{b} \,|\, \mathbf{B})\mathbf{s} = (\mathbf{B}\mathbf{t} + \mathbf{e} \,|\, \mathbf{B})\begin{pmatrix} 1 \\ -\mathbf{t} \end{pmatrix} = \mathbf{e}.$$
  4) Return $\mathsf{sk} \leftarrow \mathbf{s}$ and $\mathsf{pk} \leftarrow \mathbf{A}$.
- $\mathbf{C} \leftarrow \mathsf{GSW.Enc}(\mathsf{params}, \mathsf{pk}, \mu)$: In order to encrypt one-bit messages $\mu \in \{0,1\}$:
  1) Let $\mathbf{G}$ be the $(n+1) \times N$ gadget matrix as above;
  2) Sample uniformly a matrix $\mathbf{R} \leftarrow \{0,1\}^{m\times N}$;
  3) Compute $\mathbf{C} = \mu\mathbf{G} + \mathbf{A}^T\mathbf{R} \pmod q \in \mathbb{Z}_q^{(n+1)\times N}$;
     In the original GSW paper this was written as $\mathsf{Flatten}(\mu\mathbf{I} + \mathsf{BitDecomp}(\mathbf{R}\mathbf{A})) \in \{0,1\}^{N\times N}$ where $\mathbf{I}$ is an identity matrix.

- $\mu' \leftarrow \mathsf{GSW.Dec}(\mathsf{params}, \mathsf{sk}, \mathbf{C})$:
  1) We have $\mathsf{sk} = \mathbf{s} \in \mathbb{Z}_q^{n+1}$;
  2) Let I be such that $q/4 < 2^{I-1} \leq q/2$. Let $\mathbf{C}_I$ be the Ith column of $\mathbf{C}$;
  3) Compute $x \leftarrow \langle \mathbf{C}_I, \mathbf{s} \rangle \pmod q$ in the range $(-q/2, q/2]$;
     Note that $\langle \mathbf{C}_I, \mathbf{s} \rangle = \mathbf{C}_I^T \mathbf{s}$ and that
     $$\mathbf{C}^T\mathbf{s} = \mu \mathbf{G}^T\mathbf{s} + \mathbf{R}^T\mathbf{A}\mathbf{s} = \mu(1, 2, 4, \ldots)^T + \mathbf{R}^T\mathbf{e},$$
     and selecting the Ith column of $\mathbf{C}$ corresponds to selecting the Ith coordinate of this vector, which is $\mu 2^{I-1} + \mathbf{R}_I^T\mathbf{e}$.
  4) Output $\mu' = ||\lfloor x/2^{I-1} \rceil||$.
     So if $|x| < 2^{I-2} \leq q/4$ then return 0 and if $|x| > 2^{I-2}$ then return 1.
- $\mathsf{GSW.Eval}(\mathsf{params}, \mathbf{C}_1, \ldots, \mathbf{C}_l)$:
- $\mathsf{GSW.Add}(\mathbf{C}_1, \mathbf{C}_2)$: output
  $$\mathbf{C}_1 + \mathbf{C}_2 = (\mu_1 + \mu_2)\mathbf{G} + \mathbf{A}^T(\mathbf{R}_1 + \mathbf{R}_2) \in \mathbb{Z}_q^{(n+1)\times N};$$
- $\mathsf{GSW.Mult}(\mathbf{C}_1, \mathbf{C}_2)$: Compute and output
  $$\begin{aligned}\mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2) &= \big(\mu_1\mathbf{G} + \mathbf{A}^T\mathbf{R}_1\big)\mathbf{G}^{-1}(\mathbf{C}_2)\\ &= \mu_1\mathbf{C}_2 + \mathbf{A}^T\mathbf{R}_1\mathbf{G}^{-1}(\mathbf{C}_2)\\ &= \mathbf{A}^T\big(\mathbf{R}_1\mathbf{G}^{-1}(\mathbf{C}_2) + \mu_1\mathbf{R}_2\big)\\ &\quad + \mu_1\mu_2\mathbf{G} \mod q.\end{aligned}$$

Note that $\mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2) \in \mathbb{Z}_q^{(n+1)\times N}$. Most notably, one may also compute a homomorphic NAND gate by outputting $\mathbf{G} - \mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2)$.

Remark 3.1. Note that the formulation of the decryption algorithm in [10] is to choose an appropriate vector $\mathbf{w}$ and compute $\mathbf{s}\mathbf{C}\mathbf{G}^{-1}(\mathbf{w}^T)$. This is considerably less efficient than the original GSW decryption algorithm (both in terms of computation time and also the size of the error term). Hence we employ the GSW decryption algorithm for our scheme.

There is also a variant of the scheme that handles messages in $\mathbb{Z}_q$ when q is a power of two. We refer the reader to find the details from [6].

3.2 Security

A sketch proof is given in [6] of the following theorem.

Theorem 3.2. *Let (n, q, χ) be the public parameter such that the $\mathsf{LWE}_{(n,q,\chi)}$ assumption holds and let $m = O(n\log(q))$. Then we can say the GSW scheme is IND-CPA secure.*

The main step in the proof is showing that $(\mathbf{A}, \mathbf{RA})$ is computationally indistinguishable from uniform.

3.3 Multi-Bit GSW Scheme (MGSW Scheme)

We now describe our MGSW scheme and its properties: a multi-bit FHE scheme with *flexibility decryption* and *one-time decryption* that is based on the hardness of LWE assumption. Given security parameter λ, we set t be the number of secret key, and we can encrypt $t \times t$ bits message at one-time.

Let $q = q(\lambda)$ be an integer and let $\chi = \chi(\lambda)$ be a distribution ensemble over $\mathbb{Z}$. The variant of GSW scheme is defined similarly to the cryptosystems proposed in [7], [10], [35]. In more detail:

- $\mathsf{params} \leftarrow \mathsf{MGSW.Setup}(1^\lambda, 1^L)$:
  1) Specially, first choose the modulus $q = q(\lambda)$, the lattice dimension parameter $n = n(\lambda, L)$. Then, in order to achieve at least 2^λ security against known LWE attacks, we choose the error distribution $\chi = \chi(\lambda, L)$ appropriately. Finally, choose parameter $m = m(\lambda, L) = O(n\log q)$ and a parameter $t = O(\log(n))$;
  2) Let $l = \lfloor \log q \rfloor + 1$ and $N = (n + t) \cdot l$ and output $\mathsf{params} = (n, q, \chi, m, t)$.
- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{MGSW.KeyGen}(\mathsf{params})$:
  1) For $i \in [t]$, draw $\mathbf{t}_i^T = (t_{i,1}, \ldots, t_{i,n})$ from $\mathbb{Z}_q^{1\times n}$ and output $\mathsf{sk}_i := \mathbf{s}_i = (\mathbf{I}_i \mid -\mathbf{t}_i^T)^T = (\mathbf{I}_i, -\mathbf{t}_i^T)^T = (0, \ldots, 1, \ldots, 0 \mid -t_{i,1}, \ldots, -t_{i,n})^T \in \mathbb{Z}_q^{(n+t)\times 1}$, where the ith position is 1;
  2) Choose a matrix $\mathbf{B} \leftarrow \mathbb{Z}_q^{m\times n}$ uniformly and t vectors $\mathbf{e}_i \leftarrow \chi^{m\times 1}$ for $i \in [t]$, then compute $\mathbf{b}_i = \mathbf{B} \cdot \mathbf{t}_i + \mathbf{e}_i \pmod q$, and output $pk = \mathbf{P} = [\mathbf{b}_1 \mid \cdots \mid \mathbf{b}_t \mid \mathbf{B}] \in \mathbb{Z}_q^{m\times(n+t)}$, where the size of pk is $O(nm\cdot \log^2 q)$. Moreover, we observe that $\mathbf{P} \cdot \mathbf{s}_i = \mathbf{e}_i \pmod q$;
  3) Output $\mathsf{pk} \leftarrow \mathbf{P}$ and $\mathsf{sk} \leftarrow \mathbf{S} := \{\mathbf{s}_1, \ldots, \mathbf{s}_t\}$. Notably, $\mathbf{P} \cdot \mathbf{S} = [\mathbf{e}_1, \ldots, \mathbf{e}_t] \pmod q$.
- $\mathbf{C} \leftarrow \mathsf{MGSW.Enc}(\mathsf{params}, \mathsf{pk}, \mathbf{M})$:
  1) In order to encrypt t bits $\mu_i \in \{0, 1\}$ for $i \in [t]$, we first embed the t bits into a $(t \times t)$ dimension diagonal matrix $\mathbf{U} = \mathsf{diag}(\mu_{1,1}, \ldots, \mu_{t,t}) \in \{0, 1\}^{t\times t}$ where $\mu_{i,j} = 0$ for $i \neq j$ and $j \in [t]$. Hereafter, for simplicity, we abbreviate $\mu_{i,i}$ to μ_i , we call it plaintext matrix. Then, we use the plaintext matrix $\mathbf{U}$ to form the message matrix
     $$\mathbf{M} = \begin{pmatrix} \mathbf{U}_{t\times t} & \mathbf{0}_{t\times n} \\ \mathbf{0}_{n\times t} & \mathbf{E}_{n\times n} \end{pmatrix} \in \{0, 1\}^{(n+t)\times(n+t)}, \tag{2}$$
     where $\mathbf{U}$ is a random matrix and we stress that $\mathbf{E}$ is the $(n \times n)$ dimensional diagonal matrix;
  2) Then, sample a uniform matrix $\mathbf{R} \leftarrow \{0, 1\}^{m\times N}$. Compute and output the ciphertext
     $$\mathbf{C} = \mathbf{M} \cdot \mathbf{G} + \mathbf{P}^T \cdot \mathbf{R} \pmod q \in \mathbb{Z}_q^{(n+t)\times N}.$$
- $\mu_{i,j} \leftarrow \mathsf{MGSW.bitDec}(\mathsf{params}, \mathsf{sk}_i, \mathbf{C}, \mathbf{w}_j)$: *Flexility Single-Bit Decryption Algorithm* is similar to the decryption algorithm of [9]. In more detail:
  1) Suppose we want to decrypt ith row and jth column bit $\mu_{i,j}$, thus we let $\mathsf{sk}_i = \mathbf{s}_i := (\mathbf{I}_i \mid -\mathbf{t}_i^T)^T \in \mathbb{Z}_q^{(n+t)\times 1}$, and then we define a vector $\mathbf{w} \in \mathbb{Z}_q^{1\times(n+t)}$ such that the jth position is $\lceil q/2 \rceil$ and other positions are zero for $j \in [t]$;
     $$\mathbf{w}_j^T = [\underbrace{0, \ldots, \lceil q/2 \rceil_j, \ldots, 0}_{t} \mid \underbrace{0, \ldots, 0}_{n}];$$

2) For $i, j = 1$ to t, compute

$$v_{i,j} = \mathbf{s}_i^T \mathbf{C} \cdot \mathbf{G}^{-1}(\mathbf{w}_j^T) \,(\mathrm{mod}\ q) \in \mathbb{Z}_q,$$

where the inner product of $\langle \mathbf{s}_i, \mathbf{C} \rangle$ equals $\mathbf{s}_i^T \cdot \mathbf{P}^T\mathbf{R} + \mathbf{s}_i^T\mathbf{MG} = \mathbf{e}_i^T\mathbf{R} + \mathbf{s}_i^T\mathbf{MG} \,(\mathrm{mod}\ q) \in \mathbb{Z}_q^{1\times N}$;

3) Output the message $\mu_{i,j} = \left\| \lfloor \frac{v_{i,j}}{q/2} \rceil \right\| \in \{0,1\}$, where $\lfloor \cdot \rceil$ denotes the operation of rounding to the nearest integer. Hence, by construction we have that the output belongs to $\{0,1\}$.

4) Finally, repeat t^2 times, we can recover the whole message. We must stress that the bitDec algorithm is very similar to [9]'s decryption algorithm which is by recovering each elements separately.

Remark 3.3. Here it is important to remark that due to the structure of public key in our scheme, we achieve the precise decryption by adjusting the position of $\lceil q/2 \rceil$ in vector $\mathbf{w}$ dynamically. Namely that we dot-multiply $\mathbf{s}_i^T\mathbf{C}$ with $\mathbf{G}^{-1}(\mathbf{w}_j)$ and get the bit at ith row and jth column of plaintext matrix.

Considering that the single-bit decryption algorithm of MGSW scheme works as described above, we can get each bit of the message by using the bitDec decryption algorithm with the appropriate secret keys. We now present the one-time decryption algorithm of MGSW scheme, which allows us recover all the bits of the message simultaneously.

- $\mathbf{U} \leftarrow \mathsf{MGSW.Dec}(\mathsf{params}, \mathsf{sk}, \mathbf{C})$: *One-time Decryption Algorithm*. We now give the formal details:

1) First, suppose the user with one-time secret key matrix $\mathbf{S} = (\mathbf{s}_1, \ldots, \mathbf{s}_t) \in \mathbb{Z}_q^{(n+t)\times t}$ as follows:

$$\mathbf{S} := (\mathbf{s}_1, \ldots, \mathbf{s}_t) = \begin{pmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \\ -t_{1,1} & \cdots & -t_{t,1} \\ \vdots & \ddots & \vdots \\ -t_{1,n} & \cdots & -t_{t,n} \end{pmatrix};$$

Here it is important to remark that

$$\begin{aligned} \mathbf{P} \cdot \mathbf{S} &= [\mathbf{b}_1 - \mathbf{Bt}_1, \ldots, \mathbf{b}_t - \mathbf{Bt}_t] \\ &= [\mathbf{e}_1, \ldots, \mathbf{e}_t] \,(\mathrm{mod}\ q) \in \mathbb{Z}_q^{m\times t}. \end{aligned}$$

Thus, it is easy for us to get the bound of $\mathbf{P} \cdot \mathbf{S}$ less than or equal to $t|\mathbf{e}|$, i.e., $\|\mathbf{P} \cdot \mathbf{S}\| \le t|\mathbf{e}|$.

2) Define a matrix $\mathbf{W} \in \mathbb{Z}_q^{t\times(n+t)}$ as follows:

$$\mathbf{W}^T := \begin{pmatrix} \lceil q/2 \rceil & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lceil q/2 \rceil \\ 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{pmatrix} \in \{0,1\}^{(n+t)\times t};$$

3) Compute and output

$$\mathbf{V}_{i,j} = \langle \mathbf{S}, \mathbf{C} \rangle \cdot \mathbf{G}^{-1}(\mathbf{W}^T) \,(\mathrm{mod}\ q) \in \mathbb{Z}_q^{t\times t};$$

where we have that $\langle \mathbf{S}, \mathbf{C} \rangle$ over $\mathbb{Z}_q^{t\times(n+t)}$, i.e.,

$$\begin{aligned} \langle \mathbf{S}, \mathbf{C} \rangle &= \mathbf{S}^T\mathbf{P}^T\mathbf{R} + \mathbf{S}^T\mathbf{MG} \\ &= [\mathbf{e}_1, \ldots, \mathbf{e}_t]^T\mathbf{R} + \mathbf{S}^T\mathbf{MG} \,(\mathrm{mod}\ q). \end{aligned}$$

4) Lastly, armed with above results, output the whole message $\mathbf{U} = \left\| \lfloor \frac{\mathbf{V}_{i,j}}{q/2} \rceil \right\| \in \{0,1\}^{t\times t}$.

Remark 3.4. Normally, we can choose different secret key sk_i to decrypt the jth column ciphertext $\mathbf{C}_j$ for $j \in [N]$ bit-by-bit and get the ith bit message of $\mathbf{C}_j$, i.e., we can get ith row and jth column bit under ith secret key. But actually we can use secret key matrix $\mathbf{S}$ to recover the whole message which is based on the one-time decryption algorithm described above. We compute $\mathbf{V}_{i,j} = \mathbf{S}^T\mathbf{C} \cdot \mathbf{G}^{-1}(\mathbf{W}^T)$ as follows:

$$\mathbf{V}_{i,j} = \left\lceil \frac{q}{2} \right\rceil \cdot \mathbf{U} + \begin{pmatrix} \mathbf{e}_1^T\mathbf{R} \\ \vdots \\ \mathbf{e}_t^T\mathbf{R} \end{pmatrix} \cdot \mathbf{G}^{-1}(\mathbf{W}^T) \in \mathbb{Z}_q^{t\times t};$$

It is straightforward to evaluate the magnitude of noise and verify that it grows linearly when compared to the flexility single-bit decryption algorithm.

- $\mathsf{MGSW.Eval}(\mathsf{params}, \mathbf{C}_1, \ldots, \mathbf{C}_l)$: There exist two algorithms, i.e., homomoriphic addition and homomorphic multiplication. For any two plaintext matrices $\mathbf{U}_1, \mathbf{U}_2 \in \{0,1\}^{t\times t}$, we get the ciphertext $\mathbf{C}_1 = \mathbf{M}_1 \cdot \mathbf{G} + \mathbf{P}^T \cdot \mathbf{R}_1$ and $\mathbf{C}_2 = \mathbf{M}_2 \cdot \mathbf{G} + \mathbf{P}^T \cdot \mathbf{R}_2$ respectively. Therefore, it holds that:
  - $\mathsf{MGSW.Add}(\mathbf{C}_1, \mathbf{C}_2) \in \mathbb{Z}_q^{(n+t)\times N}$: output

$$\mathbf{C}_1 + \mathbf{C}_2 = (\mathbf{M}_1 + \mathbf{M}_2)\mathbf{G} + \mathbf{P}^T(\mathbf{R}_1 + \mathbf{R}_2) \,(\mathrm{mod}\ q);$$

  - $\mathsf{MGSW.Mult}(\mathbf{C}_1, \mathbf{C}_2) \in \mathbb{Z}_q^{(n+t)\times N}$: output

$$\begin{aligned} \mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2) &= \left(\mathbf{M}_1\mathbf{G} + \mathbf{P}^T\mathbf{R}_1\right) \cdot \mathbf{G}^{-1}(\mathbf{C}_2) \\ &= \mathbf{P}^T\mathbf{R}_1\mathbf{G}^{-1}(\mathbf{C}_2) + \mathbf{M}_1\mathbf{P}^T\mathbf{R}_2 \\ &\quad + \ \mathbf{M}_1\mathbf{M}_2\mathbf{G} \,(\mathrm{mod}\ q); \end{aligned} \tag{3}$$

  - $\mathsf{MGSW.NAND}(\mathbf{C}_1, \mathbf{C}_2)$: This also allows us to compute a homomorphic NAND gate by outputting $\mathbf{G} - \mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2)$.

Remark 3.5. We observe that our multi-bit GSW scheme can achieve $t \times t$ bits homomorphic addition operation, whereas only supports t bits homomorphic multiplication since the (i,j)-element in $\mathbf{U}_1 \times \mathbf{U}_2$ is not the product of $\mu_{1_{i,j}} \times \mu_{2_{i,j}}$.

3.4 Correctness Analysis

Next we will analyze the correctness of MGSW scheme.

Definition 3.6. *We say that a ciphertext* $\mathbf{C}$ *that is designed to encrypt* $\mathbf{U} \in \mathbb{Z}_q^{t\times t}$ *(see Eq. (2)), under* t *different secret keys* $\mathbf{s}_i$ *for* $i \in [t]$*. We denote the noise vector* $\mathrm{noise} \in \mathbb{Z}_q^{1\times N}$ *for flexility*

single bit decryption algorithm bitDec, *if it holds that single bit message's noise is as follows:*

$$\mathrm{noise}_{(\mathbf{s}_i,\mathbf{M})}(\mathbf{C}) = \mathbf{s}_i^T\mathbf{C} - \mathbf{s}_i^T\mathbf{M}\mathbf{G} = \mathbf{s}_i^T\mathbf{P}^T\mathbf{R} = \mathbf{e}_i^T\mathbf{R};$$

To simple notation, whenever $\mathbf{M}$ *and* $\mathbf{C}$ *are clear from context we abbreviate* $\mathrm{noise}_{(\mathbf{s}_i,\mathbf{M})}(\mathbf{C})$ *to* $\mathrm{noise}_{\mathbf{s}_i}$.

We must stress that, in our setting, due to the structure of our new public key, $\mathrm{noise}_{\mathbf{s}_i}$ is the noise of ith row of plaintext matrix $\mathbf{U}$ rather than the noise of single bit.

Corollary 3.7. *Obviously, armed with the Definition 3.6, for convenience, we can denote the whole noise matrix* $\mathrm{Noise}_{(\mathbf{S},\mathbf{M})}(\mathbf{C}) = (\mathrm{noise}_{\mathbf{s}_1},\ldots,\mathrm{noise}_{\mathbf{s}_t})^T \in \mathbb{Z}_q^{t\times N}$ *for one-time decryption algorithm* Dec, *if it holds that multiple noises*

$$\mathrm{Noise}_{(\mathbf{S},\mathbf{M})}(\mathbf{C}) = \mathbf{S}^T\cdot\mathbf{P}^T\cdot\mathbf{R} = \begin{pmatrix}\mathrm{noise}_{\mathbf{s}_1}\\ \vdots \\ \mathrm{noise}_{\mathbf{s}_t}\end{pmatrix} (\mathrm{mod}\ q),$$

where $\mathbf{S} = [\mathbf{s}_1,\ldots,\mathbf{s}_t]$ *is the one-time secret key matrix. For future convenience, whenever* $\mathbf{M}$ *and* $\mathbf{C}$ *are clear from context we abbreviate* $\mathrm{Noise}_{(\mathbf{S},\mathbf{M})}(\mathbf{C})$ *to* $\mathrm{Noise}_{\mathbf{S}}$.

In order to analyze correctness, it is convenient to define the following notion of *noisy ciphertexts*.

Definition 3.8 (E-noisy ciphertext). *An E-noisy ciphertext is a matrix* $\mathbf{C}\in\mathbb{Z}_q^{(m+1)\times N}$ *such that* $\langle\mathbf{s}_i,\mathbf{C}\rangle = \mathbf{s}_i^T\cdot\mathbf{M}\cdot\mathbf{G} + \mathbf{e}_i^T\cdot\mathbf{R}$ *for a corresponding message* $\mathbf{M}$, *and under secret key* $\mathbf{s}_i\in\mathbb{Z}_q^{(n+t)\times 1}$. *Then, we set the norm of* $\mathrm{noise}_{\mathbf{s}_i}$ *as*

$$||\mathrm{noise}_{\mathbf{s}_i}|| \le ||\mathbf{e}_i^T\mathbf{R}|| \le ||\mathbf{e}_i^T||_2\cdot||\mathbf{R}||_\infty \le \sqrt{N}\cdot 2\sqrt{m}B \le E.$$

Corollary 3.9. *For the one-time secret key matrix* $\mathbf{S}\in\mathbb{Z}_q^{(n+t)\times t}$, *when we run* Dec *algorithm, we can obtain* $\mathrm{Noise}_{\mathbf{S}} = [\mathbf{e}_1,\ldots,\mathbf{e}_t]^T\cdot\mathbf{R}$. *Thus, in this setting, we obtain*

$$\|\mathrm{Noise}_{\mathbf{S}}\| \le t\cdot\|\mathrm{noise}_{\mathbf{s}_i}\| \le t\cdot E.$$

Lemma 3.10. *We say the noise level of ciphertext* $\mathbf{C}$ *with respect to plaintext matrix* $\mathbf{U}$ *(one composition of* $\mathbf{M}$*) and secret key* $\mathbf{s}_i$ *for* $i\in[t]$, *then the noise vector holds that*

$$t\|\mathrm{noise}_{\mathbf{s}_i}\| = \|\mathrm{Noise}_{\mathbf{S}}\|.$$

Next we analyze the correctness of decryption.

Lemma 3.11. *Let* $\mathbf{C}$ *be an E-noisy encryption of* $\mathbf{M}$, *if we can recover* $\mu_{i,j}$ *(one element of* $\mathbf{U}$*) from ciphertext* $\mathbf{C}$ *under a secret key* $\mathbf{s}_i$, *then there exists*

$$\mu_{i,j} := \langle\mathbf{s}_i,\mathbf{C}\rangle\cdot\mathbf{G}^{-1}(\mathbf{w}_j^T) = (\mathrm{noise}_{\mathbf{s}_i} + \mathbf{s}_i^T\mathbf{M}\mathbf{G})\cdot\mathbf{G}^{-1}(\mathbf{w}_j^T),$$

such that $\|\mathrm{noise}_{\mathbf{s}_i}\cdot\mathbf{G}^{-1}(\mathbf{w}_j^T)\|_\infty \le \|\mathrm{noise}_{\mathbf{s}_i}\|\cdot\|\mathbf{G}^{-1}(\mathbf{w}_j^T)\| \le N\cdot E < q/8$.

Proof. Clearly, we can easily prove Lemma 3.11 by using Lemmas 3.8 and omit further details. □

Lemma 3.12. *Let* $\mathbf{C}$ *be an E-noisy encryption of* $\mathbf{M}$, *if we can recover the whole* $\mathbf{U}$ *from ciphertext* $\mathbf{C}$, *then there exists a secret key matrix* $\mathbf{S}$ *such that*

$$\mathbf{V} := \langle\mathbf{S},\mathbf{C}\rangle\cdot\mathbf{G}^{-1}(\mathbf{W}^T) = (\mathrm{Noise}_{\mathbf{S}} + \mathbf{S}^T\mathbf{M}\mathbf{G})\cdot\mathbf{G}^{-1}(\mathbf{W}^T),$$

with $\|\mathrm{Noise}_{\mathbf{S}}\cdot\mathbf{G}^{-1}(\mathbf{W}^T)\|_\infty \le N\cdot tE < q/8$.

Proof. The proof can be obtained directly from Lemmas 3.8 and 3.11. Now one can observe that decryption works correctly as long as $\|\mathrm{Noise}_{\mathbf{S}}\cdot\mathbf{G}^{-1}(\mathbf{W}^T)\|_\infty \le \frac{q}{8}$, i.e., $E < \frac{q}{4tN}$. Hence, we call this value $E = \frac{q}{4tN}$ as the bound of noise. □

Below we analyze the homomorphic operation. Before presenting the bound of noise, we first give the following remark,

Remark 3.13. For easier reading, below we set $\Upsilon_{\mathbf{C}_1} := \mathrm{Noise}_{(\mathbf{S},\mathbf{M}_1)}(\mathbf{C}_1)$ and $\Upsilon_{\mathbf{C}_2} := \mathrm{Noise}_{(\mathbf{S},\mathbf{M}_2)}(\mathbf{C}_2)$.

Lemma 3.14 ([14] Lemma 3.2). *The noise in homomorphic addition, homomorphic multiplication and homomorphic negation is bounded as follows:*

- *Addition, for* $\mathbf{M}_1,\mathbf{M}_2\in\{0,1\}^{(n+t)\times(n+t)}$, *it holds that*

  $$\|\mathrm{Noise}_{(\mathbf{S},\mathbf{M}_1+\mathbf{M}_2)}(\mathbf{C}_1+\mathbf{C}_2)\| \le \|\Upsilon_{\mathbf{C}_1}\| + \|\Upsilon_{\mathbf{C}_2}\|.$$

- *Multiplication, for* $\mathbf{M}_1,\mathbf{M}_2$, *it holds that*

  $$\|\mathrm{Noise}_{(\mathbf{S},(\mathbf{M}_1\cdot\mathbf{M}_2))}\big(\mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2)\big)\| \le \|\mathbf{U}_1\|_2\cdot\|\Upsilon_{\mathbf{C}_2}\|_\infty + \|\mathbf{G}^{-1}(\mathbf{C}_2)\|_\infty\cdot\|\Upsilon_{\mathbf{C}_1}\|_\infty.$$

- *Negation, for* $\mathbf{M}$, *it holds that*

  $$\|\mathrm{Noise}_{(\mathbf{S},\mathbf{M})}(\mathbf{G}-\mathbf{C})\| = \|\mathrm{Noise}_{(\mathbf{S},\mathbf{M})}(\mathbf{C})\|.$$

Proof. Let $\mathbf{S}\in\mathbb{Z}^{(n+t)\times t}$ be a secret key matrix. Let $\mathbf{C}_1,\ \mathbf{C}_2\in\mathbb{Z}_q^{(m+1)\times N}$ be ciphertexts that encrypt message $\mathbf{M}_1,\ \mathbf{M}_2\in\{0,1\}^{(n+t)\times(n+t)}$, respectively. Then

- Addition, the ciphertexts addition results in ciphertexts $\mathbf{C}^{\mathsf{Add}} = \mathbf{C}_1+\mathbf{C}_2\ (\mathrm{mod}\ q)$ such that

  $$\langle\mathbf{S},\mathbf{C}^{\mathsf{Add}}\rangle = \mathrm{Noise}_{(\mathbf{S},\mathbf{M}_1+\mathbf{M}_2)} + \mathbf{S}^T\cdot\mathbf{M}^{Add}\cdot\mathbf{G},$$

  where $\mathbf{M}^{\mathsf{Add}} = \mathbf{M}_1+\mathbf{M}_2$ and noise is

  $$\mathrm{Noise}_{(\mathbf{S},\mathbf{M}_1+\mathbf{M}_2)} = \mathrm{Noise}_{(\mathbf{S},\mathbf{M}_1)} + \mathrm{Noise}_{(\mathbf{S},\mathbf{M}_2)}.$$

  Clearly, its is $t\cdot(E_1+E_2)$-noisy.
- Multiplication, the ciphertexts multiplication results in ciphertext $\mathbf{C}^{\mathsf{Mult}} = \mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2)\in\mathbb{Z}_q^{(n+t)\times N}$ s.t.,

  $$\mathbf{C}^{\mathsf{Mult}} = \mathbf{M}_1\cdot\mathbf{M}_2\cdot\mathbf{G} + \big(\mathbf{P}^T\mathbf{R}_1\mathbf{G}^{-1}(\mathbf{C}_2) + \mathbf{M}_1\mathbf{P}^T\mathbf{R}_2\big),$$

  then $\langle\mathbf{S},\mathbf{C}^{\mathsf{Mult}}\rangle$ equals

  $$\mathbf{S}^T\Big(\mathbf{M}_1\mathbf{M}_2\mathbf{G} + \big(\mathbf{P}^T\mathbf{R}_1\mathbf{G}^{-1}(\mathbf{C}_2) + \mathbf{M}_1\mathbf{P}^T\mathbf{R}_2\big)\Big).$$

  For future convenience, we first set the noise as $\mathrm{Noise}_{(\mathbf{S},\mathbf{M}_1\mathbf{M}_2)} = \mathbf{S}^T\big(\mathbf{P}^T\mathbf{R}_1\mathbf{G}^{-1}(\mathbf{C}_2) + \mathbf{M}_1\mathbf{P}^T\mathbf{R}_2\big)$. Clearly, $\|\Upsilon_{\mathbf{C}_1}\| = \|\mathbf{S}^T\mathbf{P}^T\mathbf{R}_1\| \le \|[\mathbf{e}_1,\ldots,\mathbf{e}_t]^T\cdot\mathbf{R}_1\| \le tE_1$ by Lemma 3.8, and $\mathbf{C}_2$ is an $(n+t)\times N$ binary matrix ($\mathbf{G}^{-1}\in\mathbb{Z}_q^{N\times(n+t)}$). Hence, in this case, there exists $\|\mathbf{S}^T\mathbf{P}^T\mathbf{R}_1\cdot\mathbf{G}^{-1}(\mathbf{C}_2)\| \le tE_2\cdot\|\mathbf{G}^{-1}(\mathbf{C}_2)\| \le N\cdot tE_2$.

Moreover, we remark that

$$\mathbf{S}^T \cdot \left(\mathbf{M}_1\mathbf{P}^T\right) = \begin{pmatrix} \left(u_i\mathbf{b}_1^T - \mathbf{t}_i^T\mathbf{B}^T\right) \\ \vdots \\ \left(u_i\mathbf{b}_t^T - \mathbf{t}_i^T\mathbf{B}^T\right) \end{pmatrix}, \tag{4}$$

where the bound of $(\mu_i \cdot \mathbf{b}_i^T - \mathbf{t}_i^T \cdot \mathbf{B}^T)$ is $|\mathbf{e}_i^T|$. Hence, $\|\mathbf{S}^T \cdot (\mathbf{M}_1\mathbf{P}^T)\| \leq \left\|[\mathbf{e}_1^T, \ldots, \mathbf{e}_t^T]^T\right\| \leq \max_i\|\mathbf{e}_i^T\|$. In this case, we can easily get the bound $\|\Upsilon_{\mathbf{C}_2}\| := \|\mathbf{S}^T \cdot \left(\mathbf{M}_1\mathbf{P}^T\mathbf{R}_2\right)\| \leq \|\mathbf{e}_i^T\mathbf{R}\| \leq E_2$. Namely that, $\|\mathbf{U}_1\|_2 \cdot \|\Upsilon_{\mathbf{C}_2}\|_\infty \leq \sqrt{t} \cdot E_2$. Therefore, we have that $\|\mathrm{Noise}_{(\mathbf{S},(\mathbf{M}_1\cdot\mathbf{M}_2))}\left(\mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2)\right)\| \leq NtE_2 + \sqrt{t}E_2$ and the ciphertext $\mathbf{C}^{\mathsf{Mult}}$ is $\left((Nt + \sqrt{t}) \cdot E\right)$-noisy.

- Negation: the same calculation holds for NAND gates, outputs the matrix product $\mathbf{G} - \mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2)$. Consider the evaluation of a Boolean circuit of depth L consisting of NAND gates. It takes as input fresh ciphertexts, i.e., E-noisy ciphertexts, and at each level the noise is multiplied by a factor of at most $(Nt + \sqrt{t})$, i.e., the norm of error elements is increased by a factor of at most $(Nt + \sqrt{t})$. Therefore, the error elements of final ciphertext has norm bounded by $E_{\text{final}} = (Nt + \sqrt{t})^L \cdot E$.

  To ensure correctness of decryption we need $E_{\text{final}} \leq \lfloor \frac{q}{2} \rfloor/4$. In other words, we have that condition$(Nt + \sqrt{t})^L \cdot E < \lfloor \frac{q}{2} \rfloor/4$ must hold, which is guaranteed by our choice of parameters.

Completing the proof. □

3.5 IND-CPA Security Analysis

Below, we show the multi-bit GSW scheme is IND-CPA secure based on the LWE assumption by using Theorem 3.15 to show that the scheme is indistinguishable from the original GSW [6] scheme.

Theorem 3.15. *([36]) Let $m > n \in \mathbb{N}$, let $q \in \mathbb{N}$ and let χ be a discrete Gaussian distribution on $\mathbb{Z}$ such that the (n, q, χ, m)-LWE problem is hard. Let t be an integer such that $t = O(\log(n))$. Define two distributions $\mathcal{X}$ and $\mathcal{Y}$ as follows.*

- *$\mathcal{X}$ is the distribution on $m \times (t + n)$ matrices*

$$[\mathbf{b}_1| \cdots |\mathbf{b}_t|\mathbf{B}],$$

where $\mathbf{B} \in \mathbb{Z}_q^{m\times n}$ is chosen uniformly at random and where, for all $1 \leq i \leq t$,

$$\mathbf{b}_i = \mathbf{B}\mathbf{t}_i + \mathbf{e}_i \pmod q,$$

where $\mathbf{t}_i$ is sampled uniformly from $\mathbb{Z}_q^n$ and $\mathbf{e}_i$ is sampled from a discrete Gaussian distribution χ.
- *$\mathcal{Y}$ is the uniform distribution on $\mathbb{Z}_q^{m\times(t+n)}$.*

Hence, in this setting, the two distributions $\mathcal{X}$ and $\mathcal{Y}$ are computationally indistinguishable.

Theorem 3.16. *Let $params = (n, q, \chi, m, t)$ be such that the $\mathsf{LWE}_{n,q,\sigma,m}$ assumption holds and $m = O(n\log q)$. Then the* MGSW *scheme is* IND-CPA *secure.*

Proof. The proof of security consists of two steps:

- First, we apply Theorem 3.15 to show that, under the LWE assumption, the matrix $\mathbf{P} = [\mathbf{b}_1, \ldots, \mathbf{b}_t, \mathbf{B}] \in \mathbb{Z}_q^{m\times(n+t)}$ is computationally indistinguishable from a randomly chosen matrix.
- Then can use the leftover hash lemma to replace the ciphertext $\mathbf{C} = \mathbf{M}\mathbf{G} + \mathbf{P}^T\mathbf{R}$ with a uniformly random value $\mathbf{C}'$, namely that $\mathbf{P}^T \cdot \mathbf{R}$ is indistinguishable from uniform assuming the hardness of $\mathsf{LWE}_{n,q,\chi,m}$.

This completes the sketch of the proof. We refer the reader to [6] for more details. □

3.6 Bootstrapping

In order to construct unbounded fully homomorphic encryption, we adopt the bootstrapping methodology of [9]. Most importantly, we need to construct a symmetric key encryption first, in more detail:

- $\mathsf{key} \leftarrow \mathsf{KeyGen}(\mathsf{params})$: Identical to the secret key matrix of MGSW scheme, and output key matrix $\mathbf{S} \in \mathbb{Z}_q^{(n+t)\times t}$;
- $\mathbf{C} \leftarrow \mathsf{MGSW.SymEnc}(\mathsf{key}, \mathbf{M})$:
  1) Sample random matrix $\mathbf{B} \leftarrow \mathbb{Z}_q^{(n+t)l\times n}$ rather than from $\mathbb{Z}_q^{m\times n}$, and $\bar{\mathbf{e}}_i \leftarrow \chi^{(n+t)l\times 1}$;
  2) Compute $\bar{\mathbf{b}}_i = \bar{\mathbf{B}} \cdot \mathbf{t}_i + \bar{\mathbf{e}}_i \pmod q \in \mathbb{Z}_q^{(n+t)l\times 1}$ and reconstruct matrix $\bar{\mathbf{A}} = [\bar{\mathbf{b}}_1, \ldots, \bar{\mathbf{b}}_t, \bar{\mathbf{B}}]^T \in \mathbb{Z}_q^{N\times(n+t)}$;
  3) Compute and output: $\mathbf{C} = \mathbf{M} \cdot \mathbf{G} + \bar{\mathbf{A}}^T \pmod q$ for the plaintext $\mathbf{M}' \in \{0, 1\}^{t\times t}$.
- $\mathbf{M} \leftarrow \mathsf{Dec}(\mathbf{C}, \mathsf{key})$: Identical to the decryption algorithm of MGSW scheme.

Next step we need to construct key-switching procedure and then adopt the optimized bootstrapping procedure [9] to achieve "pure" FHE, where the optimized bootstrapping procedure consists of two algorithms $\mathsf{BootGen}(\cdot)$ and $\mathsf{Bootstrap}(\cdot)$ which were first proposed by Alperin-Sheriff and Peikert at Crypto'14 [7]. Actually, bootstrapping has no direct relationship with multi-bit encryption in our work, since we can adopt the methodology of [9] without any modification.

Therefore, we omit further details and just sketch the important steps.

1) $\mathsf{BootGen}(\mathbf{s} \in \mathbb{Z}_q^d, \mathsf{sk})$: Input a secret key vector $\mathbf{s} \in \mathbb{Z}_q^d$ for the ciphertexts from the scheme to be bootstrapped, and a secret key sk from our scheme MGSW, then outputs a bootstrapping key evk, which appropriately encrypts $\mathbf{s}$ under sk

$$\mathbf{P}_1, 0, \ldots, 0 \leftarrow \mathsf{SymEnc}(\mathsf{key}, \mathsf{diag}(1, 0, \ldots, 0))$$

$$\tau_{i,j} \leftarrow \mathsf{SymEnc}(\mathsf{key}, \mathsf{diag}(\phi_i(s_j))) \text{ for } i \in [t],\ j \in [d],$$

where $\phi_i(s_j) := \log \Pi_{p\leq s_j} p^{\lfloor \log_p s_j \rfloor}$ by Section 2.2 from Alperin-Sheriff and Peikert [7]. Then compute and get $\mathsf{ssk}_{i,j} \leftarrow \mathsf{SwitchKeyGen}(\mathsf{sk}, \pi_{\phi_i(s_j)})$, where $\pi_{\phi_i(s_j)}$ is the cyclic permutation that maps the $\phi_i(s_j)$, then compute and get

$$\mathsf{ssk}_{\phi_i(x)} \leftarrow \mathsf{SwitchKeyGen}(\mathsf{sk}, \pi_{\phi_i(x)}).$$

TABLE 1
A Comparison of Our Leakage-Resilient Scheme with the Leakage-Resilient HAO and BL Schemes

Ours Parameter Setting	LR-HAO Parameter	BL [8] Parameter	Note
$n = \tau^2$	$n = \tau^2$	$n = \tau^2$	lattice dimension;
$m = m(\lambda, L) \geq (n + t)$	$m = m(\lambda, L) \geq (n + t)$	$m = m(\lambda, L) \geq n$	lattice dimension;
$q \geq \tau \cdot 2^{2\tau \log^2 \tau}$	$q \geq \tau \cdot 2^{2\tau \log^2 \tau}$	$q \geq \tau \cdot 2^{2\tau \log^2 \tau}$	modulus;
$\beta = \tau \cdot \tau^{\log \tau}$	$\beta = \tau \cdot \tau^{\log \tau}$	$\beta = \tau \cdot \tau^{\log \tau}$	bounded;
$\chi = \bar{\Psi}_\beta$	$\chi = \bar{\Psi}_\beta$	$\chi = \bar{\Psi}_\beta$	error distribution;
$m \geq (n + t) \log q + 3\lambda$	$m \geq (n + t) \log q + 3\lambda$	$m \geq 2n \log q + 3\lambda$	lattice dimension;
$\eta \leq nt - 2t \log q - 4\lambda$	$\eta \leq nt - 4t \log q - 4\lambda$	$\eta \leq n - 2 \log q - 4\lambda$	leakage bits

For $\mathbf{x} \in \mathbb{Z}^r$ we denote $\mathsf{diag}(\mathbf{x}) \in \mathbb{Z}^{r \times r}$ is a diagonal matrix and there exists $\pi_{\phi_i(x)}$ which is the cyclic permutation that maps the $(x \pmod{r_i})$th row to the first row in the matrix. Finally, output the $\mathsf{evk} = (\mathbf{P}, \tau_{i,j}, \mathsf{ssk}_{i,j}, \mathsf{ssk}_{\phi_i(x),})$.

2) $\mathsf{Bootstrap}(\mathsf{evk}, \mathbf{c} \in \{0,1\}^d)$: Input a bootstrapping key evk and a ciphertext $\mathbf{c} \in \mathbb{Z}_q^d$, output the refreshed ciphertexts $\mathbf{C}'$. We omit the inner product and rounding algorithms, please find more details from [9].

Remark 3.17. As another independent and concurrent work at ITCS'2014, Brakerski and Vaikuntanathan [33] reached the major milestone of a LWE-based bootstrapping procedure. However, comparing with Alperin-Sheriff and Peikert method [7], the approach of [33] results with very large polynomial runtimes and approximate factors.

4 Leakage Resilient GSW-FHE on Multi-Bit Message (LRMGSW Scheme)

In this section, we present LRMGSW scheme, an adaptively leakage resilient variant of the Berkoff and Liu's LRGSW scheme [8] (hereafter BL). In our paper, we adopt BL's method and apply it to multi-bit FHE scheme. Obviously, this is a natural extension of leakage resilient on multi-bit scheme.

4.1 Leakage Resilient MGSW Scheme

We now present our LRMGSW scheme which is leakage resilient, it's easy to extent leakage resilient MGSW from single-bit leakage resilient leveled FHE scheme [8]. There is no difference with MGSW scheme except the Setup algorithm. In LRMGSW scheme, we also adopt BL scheme's parameters as follows Table 1. In more detail:

- $\mathsf{LRMGSW.Setup}(1^\lambda, 1^L)$: λ is a security parameter and $L = poly(\lambda)$ is the maximum circuit depth our scheme can evaluate. Most importantly, we set $\tau = \max\{L, \lambda^2\}$:
  And let $l = \lfloor \log q \rfloor + 1$ and $N = (n + t) \cdot l$ where t also is the number of bits we want to encrypt.
- $\mathsf{LRMGSW.KeyGen}(\cdot)$ identical to $\mathsf{MGSW.KeyGen}(\cdot)$, $\mathsf{LRMGSW.Enc}(\cdot)$ identical to $\mathsf{MGSW.Enc}(\cdot)$ and $\mathsf{LRMGSW.Dec}(\cdot)$ identical to $\mathsf{MGSW.Dec}(\cdot)$.

Hence, we omit further details.

4.2 Leakage Resilient MGSW (LRMFHE Scheme)

Below we present the adaptive leakage resilient of LRMGSW, we easily find MGSW scheme can be extended to LRMGSW. We achieve leakage resilient scheme by the technology of BL[8]. In this setting, a key pair $(\mathsf{pk}, \mathsf{sk})$ is first chosen by running the key generation algorithm $\mathsf{KeyGen}(1^\lambda)$ with security parameter λ, and then the adversary on input pk chooses leakage functions h_i adaptively (depending on the pk and the outputs of $h_j(\mathsf{sk})$, for $j < i$) and the adversary receives $h_i(\mathsf{sk})$. The total number of bits output by $h_i(\mathsf{sk})$ for all i.

Definition 4.1 ([8], Adaptive security). *Let η be a non-negative integer. A* LRMGSW *scheme is adaptively leakage resilient (ALR) to η bits of leakage, if for any* PPT *adversary $\mathcal{A}$, it holds that*

$$Adv_{ALR_b^\eta}(\mathcal{A}) = \big|\Pr[\mathcal{A}(ALR_{b=0}^\eta) = 1] - \Pr[\mathcal{A}(ALR_{b=1}^\eta) = 1]\big| = \mathrm{negl}(\eta), \tag{5}$$

and the experiment ALR_b^η is defined as follows:

1) *Setup: The challenger generates $(\mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_t) \leftarrow \mathsf{LRMGSW.KeyGen}(\mathsf{params})$, then keeps sk_i, $i \in [t]$ privately and sends pk to the adversary.*
2) *Pre-Challenge Leakage: The adversary $\mathcal{A}$ can adaptively ask the challenger for the following leakage query:*
   a) *Pre-Challenge Leakage queries-1: The adversary $\mathcal{A}$ can adaptively select a leakage function $h : \{0,1\}^* \rightarrow \{0,1\}^\lambda$ after viewing the pk and sends it to the challenger;*
   b) *Pre-Challenge Leakage queries-2: The challenger replies with $h(\mathsf{sk}_i)$, $i \in [t]$.*
3) *Query: The adversary $\mathcal{A}$ replies with two messages $(m_0, m_1) \in \{0,1\}^*$*
4) *Challenge: The challenger chooses a random bit $b \leftarrow \{0,1\}$, encrypts the message m_b, computes $c \leftarrow \mathsf{LRMGSW.Enc}(\mathsf{pk}, m_b)$ and sends the ciphertext c to the adversary $\mathcal{A}$;*
5) *Guess: The adversary $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$*

Berkoff and Liu [8] proved that GSW with larger parameters is bounded memory leakage. In this work, we primarily consider the adaptive bounded memory leakage model. The model is described in the works [18], [37]. Since an adversary can choose its leakage function after seeing the public key(s), in effect we consider functions that leak on the public and secret keys together. Before proving the Lemma 4.4, we first introduce the following two Lemmas.

Lemma 4.2. *(Generalized Leftover Hash Lemma, [27] Lemma 2.4 and [8] Lemma A.2) For a security parameter λ, let $n = poly(\lambda)$, let $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{m\times n}$ and $\mathbf{s} \xleftarrow{\$} \mathcal{D} \in \mathbb{Z}_q^n$, let x be any random variable, and let $k = \tilde{H}_\infty(\mathbf{s}|x)$. If there exists $m\log q \le k - 2log(\frac{1}{\varepsilon}) + 2$ then we have that $\Delta\big((\mathbf{C}, \mathbf{Cs}, x), (\mathbf{C}, \mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m, x)\big) \le \varepsilon$. In particular, setting $\varepsilon = 2^{-\lambda}$, if there exists $m\log q \le k - 2\lambda + 2$ then we have that $(\mathbf{C}, \mathbf{Cs}, x) \approx_s (\mathbf{C}, \mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m, x)$.*

Lemma 4.3. *([8] Lemma 4.2)) There exists a distribution Lossy such that $\tilde{\mathbf{A}} \leftarrow \mathsf{Lossy} \approx_c \mathbf{U} \xleftarrow{\$} \mathbb{Z}_q^{m\times n}$ and given $\mathbf{t} \xleftarrow{\$} \mathbb{Z}_q^n$, and $\mathbf{e} \leftarrow \chi^m$, $\tilde{H}_\infty^\varepsilon(\mathbf{t} \,|\, \tilde{\mathbf{A}}, \tilde{\mathbf{A}}\mathbf{t} + \mathbf{e}) \ge n$, where $\varepsilon = negl(n)$.*

In adaptive security game, the adversary's view is distribution $\mathcal{D}_{ALR} = (\mathbf{A}', \mathbf{Ct}, h(\mathbf{A}', \mathbf{s}))$, where $\mathbf{A}'$ is a public key, $\mathbf{Ct}$ is ciphertexts. In fact, suppose there exists another distribution $\mathcal{D}_{REAL} = (\mathbf{A}', \mathbf{U} \leftarrow \mathbb{Z}_q^{n\times N}, h(\mathbf{A}', \mathbf{s}))$, our goal is to show $\mathcal{D}_{ALR} \approx_c \mathcal{D}_{REAL}$. Hence, in order to show LRMGSW is secure against key leakage attack, we first prove the following lemma.

Lemma 4.4. *Assume there exits*

$$\mathcal{D}'_{origin} = \mathcal{D}_1 \approx_s \mathcal{D}_2 \approx_c \mathcal{D}_3 \approx_s \mathcal{D}_4 \approx_c \mathcal{D}_5 = \mathcal{D}_{real},$$

then we know that $\mathcal{D}'_{origin} \approx_c \mathcal{D}_{real}$, i.e., $\mathcal{D}_{ALR} \approx_c \mathcal{D}_{REAL}$.

Below, we describe these distributions in more detail:
Distribution $\mathcal{D}_{origin}$ is as follows,

$$\Big(\mathbf{b}_1, \ldots, \mathbf{b}_t, \mathbf{B} \,|\, \mathbf{b}_1^T\mathbf{R}, \ldots, \mathbf{b}_t^T\mathbf{R}, \mathbf{B}^T\mathbf{R} \,|\, h(\mathbf{B}, \mathbf{t}_1, \ldots, \mathbf{t}_t, \mathbf{e}_1, \ldots, \mathbf{e}_t)\Big).$$

Below, considering the distribution $\mathcal{D}'_{origin}$, where $\mathbf{R} := [\mathbf{r}_1, \ldots, \mathbf{r}_N] \in \{0,1\}^{m\times N}$, for $\mathbf{r}_j \in \{0,1\}^{m\times 1}$ with $j \in [N]$. Then there exist two cases need to consider:

- **1).** For $i \le j$ the element $(\mathbf{Bt}_i + \mathbf{e}_i)^T \cdot \mathbf{r}_j$ is replaced by choosing $u' \leftarrow \mathbb{Z}_q$;
- **2).** and for $i > j+1$ the element $\mathbf{B}^T \cdot \mathbf{r}_j$ is replaced by choosing $\mathbf{u} \leftarrow \mathbb{Z}_q^{n\times 1}$;

For convenience of description, we set $\mathbf{t}^* := \mathbf{t}_1, \ldots, \mathbf{t}_t$ and $\mathbf{e}^* := \mathbf{e}_1, \ldots, \mathbf{e}_t$, then we have that

$$\Big(\mathbf{b}_1, \ldots, \mathbf{b}_t, \mathbf{B} \,|\, \mathbf{b}_1^T\mathbf{r}_j, \ldots, \mathbf{b}_t^T\mathbf{r}_j, \mathbf{B}^T\mathbf{r}_j \,|\, h(\mathbf{B}, \mathbf{t}^*, \mathbf{e}^*)\Big).$$

Distribution $\mathcal{D}_1$, where $\mathbf{b}_i = \mathbf{Bt}_i + \mathbf{e}_i \in \mathbb{Z}_q^{m\times 1} \pmod q$

$$\Big(\mathbf{Bt}_1 + \mathbf{e}_1, \ldots, \mathbf{Bt}_t + \mathbf{e}_t, \mathbf{B} \,|\, \big((\mathbf{Bt}_1)^T\mathbf{r}_j + (\mathbf{e}_1^T\mathbf{r}_j)\big), \ldots, \big((\mathbf{Bt}_t)^T\mathbf{r}_j + (\mathbf{e}_t^T\mathbf{r}_j)\big), \mathbf{B}^T\mathbf{r}_j \,|\, h(\mathbf{B}, \mathbf{t}^*, \mathbf{e}^*)\Big).$$

Distribution $\mathcal{D}_2$, where $\mathbf{B}^T\mathbf{r}_j \approx_s \mathbf{u}_j \leftarrow \mathbb{Z}_q^{n\times 1}$ for $i \le j$ and $j \in [N]$, then it holds that

$$\Big(\mathbf{Bt}_1 + \mathbf{e}_1, \ldots, \mathbf{Bt}_t + \mathbf{e}_t, \mathbf{B} \,|\, \big(\mathbf{t}_1^T\boxed{\mathbf{u}_j} + (\mathbf{e}_1^T\mathbf{r}_j)\big), \ldots, \big(\mathbf{t}_t^T\boxed{\mathbf{u}_j} + (\mathbf{e}_t^T\mathbf{r}_j)\big), \boxed{\mathbf{u}_j} \,|\, h(\mathbf{B}, \mathbf{t}^*, \mathbf{e}^*)\Big).$$

Distribution $\mathcal{D}_3$, where $\tilde{\mathbf{B}} \leftarrow \mathsf{Lossy}$, it holds that

$$\Big(\tilde{\mathbf{B}}\mathbf{t}_1 + \mathbf{e}_1, \ldots, \tilde{\mathbf{B}}\mathbf{t}_t + \mathbf{e}_t, \tilde{\mathbf{B}} \,|\, \big(\mathbf{t}_1^T\boxed{\mathbf{u}_j} + (\mathbf{e}_1^T\mathbf{r}_j)\big), \ldots, \big(\mathbf{t}_t^T\boxed{\mathbf{u}_j} + (\mathbf{e}_t^T\mathbf{r}_j)\big), \boxed{\mathbf{u}_j} \,|\, h(\tilde{\mathbf{B}}, \mathbf{t}^*, \mathbf{e}^*)\Big).$$

Distribution $\mathcal{D}_4$, where $\mathbf{t}_i^T\mathbf{u}_j + \mathbf{e}_i^T\mathbf{r}_j \approx_s y_i^j \leftarrow \mathbb{Z}_q$ for $i+1 > j$, then we have that

$$\Big(\tilde{\mathbf{B}}\mathbf{t}_1 + \mathbf{e}_1, \ldots, \tilde{\mathbf{B}}\mathbf{t}_t + \mathbf{e}_t, \tilde{\mathbf{B}} \,|\, \big(y_1^j\big), \ldots, \big(y_t^j\big), \boxed{\mathbf{u}_j} \,|\, h(\tilde{\mathbf{B}}, \mathbf{t}^*, \mathbf{e}^*)\Big).$$

Distribution $\mathcal{D}_5$, where $\tilde{\mathbf{B}} \leftarrow \mathsf{Lossy}$

$$\Big(\mathbf{Bt}_1 + \mathbf{e}_1, \ldots, \mathbf{Bt}_t + \mathbf{e}_t, \mathbf{B} \,|\, \big(y_1^j\big), \ldots, \big(y_t^j\big), \boxed{\mathbf{u}_j} \,|\, h(\mathbf{B}, \mathbf{t}^*, \mathbf{e}^*)\Big).$$

Distribution $\mathcal{D}_{real}$, where $\mathbf{y}_i^j \leftarrow \mathbb{Z}_q^{1\times N}$ and $\mathbf{U} \leftarrow \mathbb{Z}_q^{n\times N}$

$$\Big(\mathbf{b}_1, \ldots, \mathbf{b}_t, \mathbf{B} \,|\, \mathbf{y}_1^j, \ldots, \mathbf{y}_t^j, \mathbf{U} \,|\, h(\mathbf{B}, \mathbf{t}^*, \mathbf{e}^*)\Big).$$

Proof. The proof of security consists of three steps(claims), as follows:

1) we argue $\mathcal{D}_1 \approx_s \mathcal{D}_2$ by Claim 4.5;
2) we argue $\mathcal{D}_3 \approx_s \mathcal{D}_4$ by Claim 4.6;
3) we argue $\mathcal{D}_2 \approx_c \mathcal{D}_3$ by Claim 4.8;

Hence, we have that: $\mathcal{D}'_{origin} = \mathcal{D}_1 \approx_s \mathcal{D}_2 \approx_c \mathcal{D}_3 \approx_s \mathcal{D}_4 \approx_c \mathcal{D}_5 = \mathcal{D}_{real}$, i.e., $\mathcal{D}_{ALR} \approx_c \mathcal{D}_{REAL}$, thus, the lemma is proved. □

Below we give rigorous proof of these claims.

Claim 4.5.

$$\mathcal{D}_1 \approx_s \mathcal{D}_2.$$

Proof. To prove the above, without loss of generality, for any random variable x, if $\tilde{H}_\infty(\mathbf{r}_j \,|\, x)$ is high enough, then $(\mathbf{B}, \mathbf{B}^T\mathbf{r}_j, x) \approx_s (\mathbf{B}, \mathbf{u}_j \leftarrow \mathbb{Z}_q^{n\times 1}, x)$, and for any t vectors $\mathbf{t}_i \leftarrow \{0,1\}^{n\times 1}$, $i \in [t]$, there exists $(\mathbf{B}, \mathbf{B}^T\mathbf{r}_j, (\mathbf{Bt}_i)^T\mathbf{r}, x) \approx_s (\mathbf{B}, \mathbf{u}_j, \mathbf{t}_i^T\mathbf{u}_j, x)$, namely,

$$(\mathbf{B}, \mathbf{B}^T\mathbf{r}_j, \underline{(\mathbf{Bt}_1)^T\mathbf{r}_j, \ldots, (\mathbf{Bt}_t)^T\mathbf{r}_j}, x) \approx_s (\mathbf{B}, \mathbf{u}_j, \underline{\mathbf{t}_1^T\mathbf{u}_j, \ldots, \mathbf{t}_t^T\mathbf{u}_j}, x).$$

Hence, we have that

$$x = \big(\mathbf{Bt}_1 + \mathbf{e}_1, \ldots, \mathbf{Bt}_t + \mathbf{e}_t, \mathbf{e}_1^T\mathbf{r}_j, \ldots, \mathbf{e}_t^T\mathbf{r}_j, h(\mathbf{B}, \mathbf{t}_1, \ldots, \mathbf{t}_t, \mathbf{e}_1, \ldots, \mathbf{e}_t)\big),$$

since $\mathrm{BitLength}(\mathbf{e}_i^T \cdot \mathbf{r}_j)$ is $l = \lfloor \log q \rfloor + 1$ bit long, and $\mathbf{r}_j$ is independent of $\mathbf{e}_i$, we can get:

$$\begin{aligned}\tilde{H}_\infty(\mathbf{r}_j \,|\, x) &\ge \tilde{H}_\infty(\mathbf{r}_j \,|\, \mathbf{e}_1^T\mathbf{r}_j, \ldots, \mathbf{e}_t^T\mathbf{r}_j, \mathbf{e}_1, \ldots, \mathbf{e}_t) \\ &\ge \tilde{H}_\infty(\mathbf{r}_j \,|\, \mathbf{e}_1, \ldots, \mathbf{e}_t) - t \cdot \mathrm{BitLength}(\mathbf{e}_i^T\mathbf{r}_j) \\ &= m - tl,\end{aligned}$$

by the Lemma 4.4, we just need to keep the inequality

$$\tilde{H}_\infty(\mathbf{r}_j \,|\, x) \ge n\log q + 2\lambda + O(1).$$

Hence, we choose $m \ge (n+t) \cdot \log q + 3\lambda$ to make the above inequality correctly. □

Claim 4.6.

$$\mathcal{D}_3 \approx_s \mathcal{D}_4.$$

Proof. There is no difference between the distribution $\mathcal{D}_3$ and $\mathcal{D}_4$ except that $\mathbf{t}_i^T\mathbf{u}_j + \mathbf{e}_i^T\mathbf{r}_j$ is replaced by $y_i \leftarrow \mathbb{Z}_q$. Thus, we can argue that: $\mathcal{D}_3' \approx_s \mathcal{D}_4'$, where the distribution $\mathcal{D}_3'$ is

$$\mathcal{D}_3' = \big(\tilde{\mathbf{B}}\mathbf{t}_1 + \mathbf{e}_1, \ldots, \tilde{\mathbf{B}}\mathbf{t}_t + \mathbf{e}_t \mid \underline{\mathbf{t}_1^T\mathbf{u}_j, \ldots, \mathbf{t}_t^T\mathbf{u}_j} \mid \mathbf{e}_1^T\mathbf{r}_j, \ldots, \mathbf{e}_t^T\mathbf{r}_j, \mathbf{u}_j, h(\tilde{\mathbf{B}}, \mathbf{t}^*, \mathbf{e}^*)\big),$$

and the distribution $\mathcal{D}_4'$ is

$$\mathcal{D}_4' = \big(\tilde{\mathbf{B}}\mathbf{t}_1 + \mathbf{e}_1, \ldots, \tilde{\mathbf{B}}\mathbf{t}_t + \mathbf{e}_t \mid \underline{v_1, \ldots, v_t}, \mathbf{e}_1^T\mathbf{r}_j, \ldots, \mathbf{e}_t^T\mathbf{r}_j, \mathbf{u}_j, h(\tilde{\mathbf{B}}, \mathbf{t}^*, \mathbf{e}^*)\big),$$

where $v_i \leftarrow \mathbb{Z}_q \ \ i \in [t]$. Actually, the adversary' view is $\mathbf{t}_i^T\mathbf{u}_j + \mathbf{e}_i^T\mathbf{r}_j$, and $\mathbf{t}_i^T\mathbf{u}_j + \mathbf{e}_i^T\mathbf{r}_j$ can be replaced by $y_i \leftarrow \mathbb{Z}_q$. However, $\mathcal{D}_3'$ contains both $\mathbf{t}_i^T\mathbf{u}_j$ and $\mathbf{e}_i^T\mathbf{r}_j$.

Now, we show the ε-smooth min-entropy of $\mathbf{t}_i \leftarrow \mathbb{Z}_q^{n\times 1}$, there exists $\varepsilon = negl(\lambda)$ such that $\forall\ \mathbf{t}_i$,

$$\begin{aligned}
&\tilde{H}_\infty^\varepsilon(\mathbf{t}_i \mid \tilde{\mathbf{B}}\mathbf{t}_1 + \mathbf{e}_1, \ldots, \tilde{\mathbf{B}}\mathbf{t}_t + \mathbf{e}_t, \tilde{\mathbf{B}}, \mathbf{e}_1^T\mathbf{r}_j, \ldots, \mathbf{e}_t^T\mathbf{r}_j, \\
&\qquad h(\tilde{\mathbf{B}}, \mathbf{t}^*, \mathbf{e}^*)) \\
&\geq \tilde{H}_\infty^\varepsilon(\mathbf{t}_i \mid \tilde{\mathbf{B}}\mathbf{t}_1 + \mathbf{e}_1, \ldots, \tilde{\mathbf{B}}\mathbf{t}_t + \mathbf{e}_t, \tilde{\mathbf{B}}) \\
&\quad - t \cdot \mathrm{BitLength}(\mathbf{e}_i^T\mathbf{r}_j) - \mathrm{BitLength}(h(\tilde{\mathbf{B}}, \mathbf{t}^*, \mathbf{e}^*)) \\
&\geq \tilde{H}_\infty^\varepsilon(\mathbf{t}_i \mid \tilde{\mathbf{B}}\mathbf{t}_1 + \mathbf{e}_1, \ldots, \tilde{\mathbf{B}}\mathbf{t}_t + \mathbf{e}_t, \tilde{\mathbf{B}}) - tl - \eta,
\end{aligned}$$

we can get $\tilde{H}_\infty^\varepsilon(\mathbf{t}_i \mid \tilde{\mathbf{B}}\mathbf{t}_1 + \mathbf{e}_1, \ldots, \tilde{\mathbf{B}}\mathbf{t}_t + \mathbf{e}_t, \tilde{\mathbf{B}}) \geq n$ by the generalized Lemma 4.3. Hence, in order to get

$$\tilde{H}_\infty^\varepsilon(\mathbf{t}_i \mid \tilde{\mathbf{B}}\mathbf{t}_1 + \mathbf{e}_1, \ldots, \tilde{\mathbf{B}}\mathbf{t}_t + \mathbf{e}_t, \tilde{\mathbf{B}}) - tl - \eta \geq \log q + 2\lambda + O(1),$$

we need $n - tl - \eta$ to be high enough. Hence, we set leakage function h to leak at most $\eta \leq n - (t+1)\log q - 4\lambda$. □

Remark 4.7. As discussed above, we just discussed leakage resilient on single-bit decryption. Actually, for leakage resilient on one-time decryption, since $\mathbf{t}_i \leftarrow \mathbb{Z}_q^{n\times 1}$ with $i \in [t]$, we set $\mathbf{T} := [\mathbf{t}_1, \ldots, \mathbf{t}_t] \in \mathbb{Z}_q^{n\times t}$, $\mathbf{e}_i \leftarrow \chi^{m\times 1}$ with $i \in [t]$, and set $\mathbf{E} := [\mathbf{e}_1, \ldots, \mathbf{e}_t] \in \chi^{m\times t}$, there exists

$$\begin{aligned}
&\tilde{H}_\infty^\varepsilon(\mathbf{T} \mid \tilde{\mathbf{B}}\mathbf{T} + \mathbf{E}, \tilde{\mathbf{B}}, \mathbf{E}^T\mathbf{r}_j, h(\tilde{\mathbf{B}}, \mathbf{T}, \mathbf{E})) \\
&\quad \geq \tilde{H}_\infty^\varepsilon(\mathbf{T} \mid \tilde{\mathbf{B}}\mathbf{T} + \mathbf{E}, \tilde{\mathbf{B}}) - \mathrm{BitLength}(\mathbf{E}^T\mathbf{r}_j) \\
&\quad - \mathrm{BitLength}(h(\tilde{\mathbf{B}}, \mathbf{T}, \mathbf{E})) \\
&\quad \geq \tilde{H}_\infty^\varepsilon(\mathbf{T} \mid \tilde{\mathbf{B}}\mathbf{T} + \mathbf{E}, \tilde{\mathbf{B}}) - tl - \eta,
\end{aligned}$$

we can get $\tilde{H}_\infty^\varepsilon(\mathbf{T} \mid \tilde{\mathbf{B}}\mathbf{T} + \mathbf{E}, \tilde{\mathbf{B}}) \geq nt$ by the generalized Lemma 4.3. Hence, in order to get

$$\tilde{H}_\infty^\varepsilon(\mathbf{T} \mid \tilde{\mathbf{B}}\mathbf{T} + \mathbf{E}, \tilde{\mathbf{B}}) - tl - \eta \geq \sum_{i\in[t]} \log q + 2\lambda + O(1).$$

We set the leakage function h to leak at most $\eta \leq nt - 2t\log q - 4\lambda$ bits.

Claim 4.8.

$$\mathcal{D}_2 \approx_c \mathcal{D}_3.$$

Proof. The claim can be easily proved by Lemma 4.3. Actually, we adopt the same methodology with BL scheme. Hence, we omit further details and recommend the reader refer to [8] for more details. □

From these Claims as described above, we can conclude the crux theorem of our paper as follows,

Theorem 4.9. *The* LRMGSW *scheme is resilient to adaptive bounded leakage of η bits, where $\eta \leq nt - 2t\log q - 4\lambda$.*

5 Leakage Resilient on HAO Scheme

In this section, we analyze the leakage resilient on HAO scheme [9] and adopt the same parameters as BL scheme in Table 1. First, we review the scheme of Hiromasa et al. [9]. We remark that, we just focus on leveled HAO scheme.

5.1 Leveled HAO Scheme

The parameters of HAO are the same as ours scheme, so we let $l := \lceil \log q \rceil$, $m := O((n+t)\log q)$, and $N := (n+t)\cdot l$. Notably, let t be the number of bits to be encrypted.

- $\mathsf{KeyGen}(1^\lambda, t)$:
  1) Set the parameters n, q, l, N, t and χ as described above with our scheme;
  2) Sample a uniformly random matrix $\mathbf{A} \xleftarrow{U} \mathbb{Z}_q^{n\times m}$, secret key matrix $\mathbf{S}' \leftarrow \chi^{t\times n}$, and noise matrix $\mathbf{E} \leftarrow \chi^{t\times m}$. Let $\mathbf{S} := [\mathbf{I}_t \mid -\mathbf{S}'] \in \mathbb{Z}_q^{t\times(n+t)}$. We denote by $\mathbf{s}_i^T$ the ith row of $\mathbf{S}$. Set
  $$\mathbf{B} := \begin{pmatrix} \mathbf{S}'\cdot\mathbf{A} + \mathbf{E} \\ \mathbf{A} \end{pmatrix} \in \mathbb{Z}_q^{(n+t)\times m}.$$
  Here it is important to remark $\mathbf{S}\cdot\mathbf{B} = \mathbf{E}$.
  3) Let $\mathbf{M}_{(i,j)} \in \{0,1\}^{t\times t}$ be the matrix with 1 in the (i,j)th position and 0 in the others. For all $i \in [m], j \in [N]$, first, we draw $\mathbf{R}_{(i,j)} \xleftarrow{U} \{0,1\}^{m\times N}$, then we set the matrix $\mathbf{P}$ as follows:
  $$\mathbf{P}_{(i,j)} := \mathbf{B}\cdot\mathbf{R}_{(i,j)} + \begin{pmatrix} \mathbf{M}_{(i,j)}\cdot\mathbf{S} \\ \mathbf{0} \end{pmatrix}\cdot\mathbf{G} \in \mathbb{Z}_q^{(n+t)\cdot N}.$$
  4) $\mathsf{pk} := (\{\mathbf{P}_{(i,j)}\}_{i,j\in[t]}, \mathbf{B})$ and $\mathsf{sk} := \mathbf{S}$.
- $\mathsf{SymEnc}_{\mathsf{sk}}(\mathbf{M} \in \{0,1\}^{t\times t})$:
  1) Sample a random matrices $\mathbf{A}' \xleftarrow{U} \mathbb{Z}_q^{n\times N}$ and $\mathbf{E} \xleftarrow{R} \chi^{t\times N}$, parse $\mathbf{S} = [\mathbf{I}_r \mid -\mathbf{S}']$,
  2) Output the ciphertext
  $$\mathbf{C} := \left[\begin{pmatrix} \mathbf{S}'\cdot\mathbf{A}' + \mathbf{E} \\ \mathbf{A}' \end{pmatrix} + \begin{pmatrix} \mathbf{M}\cdot\mathbf{S} \\ \mathbf{0} \end{pmatrix}\cdot\mathbf{G}\right]_q \in \mathbb{Z}_q^{(n+t)\times N}.$$
- $\mathsf{PubEnc}(\mathsf{pk}, \mathbf{M} \in \{0,1\}^{t\times t})$:
  1) Sample a random matrix $\mathbf{R} \xleftarrow{U} \{0,1\}^{m\times N}$;
  2) Output the ciphertext
  $$\mathbf{C} := \mathbf{B}\cdot\mathbf{R} + \sum_{i,j\in[t]:\mathbf{M}_{[i,j]}=1} \mathbf{P}_{(i,j)} \in \mathbb{Z}_q^{(n+t)\times N},$$
  where $\mathbf{M}[i,j]$ is the (i,j)th element of $\mathbf{M}$. Moreover, most notably, in the following, we will omit the subscript $\sum_{i,j\in[t]:\mathbf{M}_{[i,j]}=1}$ for the sake of clarity.

- $\mathsf{Dec}(\mathsf{sk}, \mathbf{C})$: Output the multi-bit message $\mathbf{M} = (\lfloor \langle \mathbf{s}_i, \mathbf{c}_{jl-1} \rangle \rfloor_2)_{i,j \in [r]} \in \{0,1\}^{t \times t}$. Here we need to remark that the decryption algorithm also achieve the "flexibility decryption" as our MGSW scheme.
- $\mathsf{Add}(\mathbf{C}_1, \mathbf{C}_2)$: Output $\mathbf{C}^{\mathsf{Add}} := \mathbf{C}_1 + \mathbf{C}_2 \in \mathbb{Z}_q^{(n+t)\times N}$ as the result of homomorphic addition between the input ciphertexts $\mathbf{C}_1$ and $\mathbf{C}_2$.
- $\mathsf{Mult}(\mathbf{C}_1, \mathbf{C}_2)$: $\mathbf{C}^{\mathsf{Mult}} := \mathbf{C}_1 \cdot \mathbf{G}^{-1}(\mathbf{C}_2) \in \mathbb{Z}_q^{(n+t)\times N}$ as the result of homomorphic multiplication between the input ciphertexts $\mathbf{C}_1$ and $\mathbf{C}_2$.

5.2 Leakage Resilient on HAO Scheme

Below we show the "leakage resilient" on Hiromasa-Abe-Okamoto (HAO) scheme, and adopt the same methodology with our LRMGSW scheme. We also just consider the distribution $(\mathsf{PK}, \mathsf{CT}, h(\mathbf{A}, t^*, e^*))$. The distribution $\mathcal{D}_{origin}$ is as follows,

$$\Big(\mathbf{P}_{(i,j)} := \mathbf{BR}_{(i,j)} + \left(\frac{\mathbf{M}_{(i,j)}\mathbf{S}}{\mathbf{0}}\right)\tilde{\mathbf{G}}, \mathbf{B} \mid \mathbf{C} := \mathbf{BR} + \sum \mathbf{P}_{(i,j)} \mid h(\mathbf{A}, t^*, e^*)\Big).$$

For easier reading of leakage function, where we denote $t^* := \mathbf{t}_1, \ldots, \mathbf{t}_t$ and $e^* := \mathbf{e}_1, \ldots, \mathbf{e}_t$. Apparently, the distribution $\mathcal{D}_{origin}$ is identical to distribution $\mathcal{D}_1$ i.e., $(\mathcal{D}_{origin} = \mathcal{D}_1)$, the distribution $\mathcal{D}_1$ is as follows,

$$\Bigg(\left(\frac{\mathbf{S}'\mathbf{A}+\mathbf{E}}{\mathbf{A}}\right)\cdot\mathbf{R}_{(i,j)} + \left(\frac{\mathbf{M}_{(i,j)}\mathbf{S}}{\mathbf{0}}\right)\mathbf{G}, \left(\frac{\mathbf{S}'\mathbf{A}+\mathbf{E}}{\mathbf{A}}\right) \mid \left(\frac{\mathbf{S}'\mathbf{A}+\mathbf{E}}{\mathbf{A}}\right)\cdot\mathbf{R} + \sum\left(\left(\frac{\mathbf{S}'\mathbf{A}+\mathbf{E}}{\mathbf{A}}\right)\cdot\mathbf{R}_{(i,j)} + \left(\frac{\mathbf{M}_{(i,j)}\mathbf{S}}{\mathbf{0}}\right)\mathbf{G}\right) \mid h(\mathbf{A}, t^*, e^*)\Bigg).$$

Apparently, the distribution $\mathcal{D}_1$ is identical to distribution $\mathcal{D}_2$ i.e., $(\mathcal{D}_1 = \mathcal{D}_2)$, the distribution $\mathcal{D}_2$ is as follows,

$$\Bigg((\mathbf{S}'\mathbf{A}+\mathbf{E})\cdot\mathbf{R}_{(i,j)}, \mathbf{A}\cdot\mathbf{R}_{(i,j)}, \mathbf{M}_{(i,j)}\mathbf{SG}, \mathbf{S}'\mathbf{A}+\mathbf{E}, \mathbf{A} \mid (\mathbf{S}'\mathbf{A}+\mathbf{E})\cdot\mathbf{R}, \mathbf{A}\cdot\mathbf{R}, \sum\left(\left(\frac{\mathbf{S}'\mathbf{A}+\mathbf{E}}{\mathbf{A}}\right)\cdot\mathbf{R}_{(i,j)} + \left(\frac{\mathbf{M}_{(i,j)}\mathbf{S}}{\mathbf{0}}\right)\mathbf{G}\right) \mid h(\mathbf{A}, t^*, e^*)\Bigg).$$

Apparently, the distribution $\mathcal{D}_2$ is identical to distribution $\mathcal{D}_3$, i.e., $(\mathcal{D}_2 = \mathcal{D}_3)$, for future convenience, we denote

- $\mathbf{s}_i' \leftarrow \chi^{1\times n}$ is ith row of $\mathbf{S}' \leftarrow \chi^{t\times n}$ and $\mathbf{s}_i \in \mathbb{Z}_q^{1\times(n+t)}$ is ith row of $\mathbf{S} = [\mathbf{I} \mid -\mathbf{S}'] \in \mathbb{Z}_q^{t\times(t+n)}$;
- $\mathbf{r}_j \in \{0,1\}^{m\times 1}$ as jth column of the matrix $\mathbf{R} \in \{0,1\}^{m\times N}$, where each $\mathbf{r}_j$ is independent binary vector for $j \in [N]$, for $i \le j$ the element $(\mathbf{S}'\mathbf{A}+\mathbf{E})\cdot\mathbf{r}_j$ is replaced by choosing $\mathbf{u}' \leftarrow \mathbb{Z}_q^{t\times 1}$, and for $i > j+1$ the element $\mathbf{A}\mathbf{r}_j$ is replaced by choosing $\mathbf{u} \leftarrow \mathbb{Z}_q^{n\times 1}$,

Hence, there exists one row of cihphertexts as follows:

$$\sum_{i,j\in[r]:\mathbf{M}[i,j]=1}\left(\left(\frac{\mathbf{S}'\mathbf{A}+\mathbf{E}}{\mathbf{A}}\right)\cdot\mathbf{r}_j + \left(\frac{\mathbf{M}_{(i,j)}\mathbf{S}}{\mathbf{0}}\right)\mathbf{g}\otimes\mathbf{I}_{1\times(n+t)}\right),$$

In this setting, for future convenience, we set $\mathbf{g}\otimes\mathbf{I}_{1\times(n+t)}$ as $\mathbf{G}'$, then the distribution $\mathcal{D}_3$ is as follows:

$$\Big((\mathbf{S}'\underline{\mathbf{A}\cdot\mathbf{r}_j}+\mathbf{E}\cdot\mathbf{r}_j), \underline{\mathbf{A}\cdot\mathbf{r}_j}, \mathbf{M}_{(i,j)}\mathbf{SG}', \mathbf{S}'\mathbf{A}+\mathbf{E}, \mathbf{A} \mid (\mathbf{S}'\underline{\mathbf{A}\cdot\mathbf{r}_j}+\mathbf{E}\cdot\mathbf{r}_j), \underline{\mathbf{A}\cdot\mathbf{r}_j}, \sum\left(\mathbf{S}'\underline{\mathbf{A}\cdot\mathbf{r}_j}+\mathbf{E}\cdot\mathbf{r}_j\right), \sum\left(\underline{\mathbf{A}\cdot\mathbf{r}_j}\right), \sum\left((\mathbf{M}_{(i,j)}\mathbf{S})\cdot\mathbf{G}'\right) \mid h(\mathbf{A}, t^*, e^*)\Big).$$

Claim 5.1. The distribution $\mathcal{D}_3$ is statistically indistinguishable from the distribution $\mathcal{D}_4$, i.e., $(\mathcal{D}_4 \approx_s \mathcal{D}_3)$. Hence, for $i > j+1$ choose $\mathbf{u}_j \leftarrow \mathbb{Z}_q^{n\times 1}$ to replace $\mathbf{A}\cdot\mathbf{r}_j$, there exits

$$\Big((\mathbf{S}'\underline{\mathbf{u}_j}+\mathbf{E}\cdot\mathbf{r}_j), \underline{\mathbf{u}_j}, \mathbf{M}_{(i,j)}\mathbf{SG}', \mathbf{S}'\mathbf{A}+\mathbf{E}, \mathbf{A} \mid (\mathbf{S}'\underline{\mathbf{u}_j}+\mathbf{E}\cdot\mathbf{r}_j), \underline{\mathbf{u}_j}, \sum\left(\mathbf{S}'\underline{\mathbf{u}_j}+\mathbf{E}\cdot\mathbf{r}_j\right), \sum\left(\underline{\mathbf{u}_j}\right), \sum\left((\mathbf{M}_{(i,j)}\mathbf{S})\cdot\mathbf{G}'\right) \mid h(\mathbf{A}, t^*, e^*)\Big) \quad (6)$$

Proof. To prove the above lemma, we consider $(\mathbf{A}\cdot\mathbf{r}_j, \mathbf{A}, x) \approx_s (\mathbf{u}_j, \mathbf{A}, x)$, then for any $\mathbf{S}' \leftarrow \chi^{t\times n}$, by leftover hash Lemma 4.2, there exists

$$(\mathbf{S}'\mathbf{A}\cdot\mathbf{r}_j, \mathbf{A}\cdot\mathbf{r}_j, \mathbf{A}, x) \approx_s (\mathbf{S}'\mathbf{u_j}, \mathbf{u}_j, \mathbf{A}, x).$$

Moreover, assume there exists

$$x := \Big((\mathbf{E}\cdot\mathbf{r}_j), \mathbf{M}_{(i,j)}\mathbf{SG}', \mathbf{S}'\mathbf{A}+\mathbf{E}, \mid (\mathbf{E}\cdot\mathbf{r}_j), \sum\left(\mathbf{E}\cdot\mathbf{r}_j\right), \sum\left((\mathbf{M}_{(i,j)}\mathbf{S})\cdot\mathbf{G}'\right) \mid h(\mathbf{A}, t^*, e^*)\Big),$$

then by the Definition 2.6 we have that

$$\begin{aligned}\tilde{H}_\infty(\mathbf{r}_j \mid x) &\ge \tilde{H}_\infty(\mathbf{r}_j \mid (\mathbf{E}\cdot\mathbf{r}_j), (\mathbf{E}\cdot\mathbf{r}_j), \sum(\mathbf{E}\cdot\mathbf{r}_j)) \\ &\ge \tilde{H}_\infty(\mathbf{r}_j \mid \mathbf{E}) - 3\cdot\mathrm{BitLength}(\mathbf{E}\cdot\mathbf{r}_j) \\ &\ge \tilde{H}_\infty(\mathbf{r}_j \mid \mathbf{E}) - 3\cdot(t\cdot l) \\ &\ge m - (N+2)tl.\end{aligned}$$

In this setting, by the generalized form of the leftover hash Lemma 4.4, we need

$$\tilde{H}_\infty(\mathbf{r}_j \mid x) \ge n\log q + 2\lambda + O(1).$$

Hence, we choose $m \ge (n+3tl)\cdot\log q + 3\lambda$. □

Claim 5.2. The distribution $\mathcal{D}_4$ is computationally indistinguishable from the distribution $\mathcal{D}_5$, i.e., $(\mathcal{D}_5 \approx_c \mathcal{D}_4)$, where $\tilde{\mathbf{A}} \leftarrow \mathsf{Lossy}$, hence, there exists distribution $\mathcal{D}_5$ as follows:

$$\Big((\mathbf{S}'\underline{\mathbf{u}_j}+\mathbf{E}\cdot\mathbf{r}_j), \underline{\mathbf{u}_j}, \mathbf{M}_{(i,j)}\mathbf{SG}', \mathbf{S}'\tilde{\mathbf{A}}+\mathbf{E}, \tilde{\mathbf{A}} \mid (\mathbf{S}'\underline{\mathbf{u}_j}+\mathbf{E}\cdot\mathbf{r}_j), \underline{\mathbf{u}_j}, \sum\left(\mathbf{S}'\underline{\mathbf{u}_j}+\mathbf{E}\cdot\mathbf{r}_j\right), \sum\left(\underline{\mathbf{u}_j}\right), \sum\left((\mathbf{M}_{(i,j)}\mathbf{S})\cdot\mathbf{G}'\right) \mid h(\tilde{\mathbf{A}}, t^*, e^*)\Big).$$

Proof. The above Claim 5.2 can be easy to prove by the Lemma 4.3. We omit further details. □

Claim 5.3. The distribution $\mathcal{D}_5$ is statistically indistinguishable from the distribution $\mathcal{D}_6$, i.e., $(\mathcal{D}_6 \approx_s \mathcal{D}_5)$, where $(\mathbf{S}'\mathbf{A}+\mathbf{E})\cdot\mathbf{r}_j \approx_s \mathbf{u}_j' \leftarrow \mathbb{Z}_q^{t\times 1}$. Hence there exists $\mathcal{D}_6$

TABLE 2
A Comparison of Our Scheme with Foremost LWE-Based Multi-Bit FHE Schemes

Scheme	BGH [16]	HAO [9]	Ours-MGSW
Assumption	LWE	LWE	LWE
$\|\|$plaintext$\|\|$	t	t	t
$\|\|$pk$\|\|$	$(n+t)t\log^2 q$	$(n+t)m\log^2 q$	$(n+t)m\log^2 q$
$\|\|$sk$\|\|$	$(n+t)t\log^2 q$	$(n+t)t\log^2 q$	$(n+t)t\log^2 q$
$\|\|$Eval key$\|\|$	$(n+t)^3\log^2 q$	$\times$	$\times$
$\|\|$ciphertext$\|\|$	$(n+t)\log q$	$(n+t)N\log q$	$(n+t)N\log q$
key tools	#	§	Δ & §
one-time dec	$\surd$	$\times$	$\surd$
flexibility dec	$\times$	$\surd$	$\surd$
leakage resilient	$\times$	We achieved	$\surd$

- #:*packed ciphertext (the methodology of [21]);*
- §: *packing message;* - Δ: pk *with* t LWE *instances;*

$$\Big((\mathbf{u}'_j), \underline{\mathbf{u}_j}, \mathbf{M}_{(i,j)}\mathbf{SG}', \mathbf{S}'\tilde{\mathbf{A}} + \mathbf{E}, \tilde{\mathbf{A}} \mid (\mathbf{u}'_j), \underline{\mathbf{u}_j}, \sum(\mathbf{u}'_j), \sum(\underline{\mathbf{u}_j}), \sum((\mathbf{M}_{(i,j)}\mathbf{S})\cdot\mathbf{G}') \mid h(\tilde{\mathbf{A}}, t^*, e^*)\Big).$$

Proof. Note that the distribution $\mathcal{D}_5$ contains $(\mathbf{S}'\underline{\mathbf{u}_j} + \mathbf{E}\cdot\mathbf{r}_j)$, actually, the adversary only sees $(\mathbf{S}'\underline{\mathbf{u}_j} + \mathbf{E}\cdot\underline{\mathbf{r}_j})$, but in order to prove that $\mathbf{S}'\underline{\mathbf{u}_j}$ can be replaced by $\mathbf{v} \leftarrow \mathbb{Z}_q^{t\times 1}$ implies that in the adversary's view, $(\mathbf{S}'\underline{\mathbf{u}_j} + \mathbf{E}\cdot\mathbf{r}_j)$ can be replaced by $\mathbf{u}' \leftarrow \mathbb{Z}_q^{t\times 1}$. Moreover, there exists $\varepsilon = negl(\lambda)$ such that we bound the ε-smooth min-entropy of $\mathbf{S}'$,

$$\begin{aligned}
&\tilde{H}_\infty^\varepsilon(\mathbf{S}' \mid \mathbf{S}'\tilde{\mathbf{A}} + \mathbf{E}, \tilde{\mathbf{A}}, \mathbf{Er}_j, \mathbf{Er}_j, \textstyle\sum(\mathbf{Er}_j), h(\tilde{\mathbf{A}}, t^*, e^*))\\
&\geq \tilde{H}_\infty^\varepsilon(\mathbf{S}' \mid \mathbf{S}'\tilde{\mathbf{A}} + \mathbf{E}, \tilde{\mathbf{A}}) - \mathrm{BitLength}(h(\tilde{\mathbf{A}}, t^*, e^*))\\
&- \mathrm{BitLength}(\mathbf{Er}_j, \mathbf{Er}_j, \textstyle\sum(\mathbf{E}\cdot\mathbf{r}_j))\\
&\geq \tilde{H}_\infty^\varepsilon(\mathbf{S}' \mid \mathbf{S}'\tilde{\mathbf{A}} + \mathbf{E}, \tilde{\mathbf{A}}) - 3tl - \eta.
\end{aligned}$$

By Lemma 4.3, we know that $\tilde{H}_\infty^\varepsilon(\mathbf{S}' \mid \mathbf{S}'\tilde{\mathbf{A}} + \mathbf{E}, \tilde{\mathbf{A}}) \geq tn$. Hence, in order to get

$$\tilde{H}_\infty^\varepsilon(\mathbf{S}' \mid \mathbf{S}'\tilde{\mathbf{A}} + \mathbf{E}, \tilde{\mathbf{A}}) - 3tl - \eta \geq t\cdot\log q + 2\lambda + O(1),$$

we need $tn - 3tl - \eta$ to be high enough. Hence, the leakage function h leaks at most $\eta \leq nt - 4t\log q - 4\lambda$. □

Claim 5.4. The distribution $\mathcal{D}_6$ is computationally indistinguishable from the distribution $\mathcal{D}_7$, i.e., $(\mathcal{D}_7 \approx_c \mathcal{D}_6)$, where $\tilde{\mathbf{A}} \leftarrow$ Lossy, there exists distribution $\mathcal{D}_7$ as follows:

$$\Big((\mathbf{u}'_j), \underline{\mathbf{u}_j}, \mathbf{M}_{(i,j)}\mathbf{SG}, \mathbf{S}'\mathbf{A} + \mathbf{E}, \mathbf{A} \mid (\mathbf{u}'_j), \underline{\mathbf{u}_j}, \sum(\mathbf{u}'_j), \sum(\underline{\mathbf{u}_j}), \sum((\mathbf{M}_{(i,j)}\mathbf{S})\cdot\mathbf{G}) \mid h(\mathbf{A}, t^*, e^*)\Big).$$

Proof. The above Claim 5.4 can be easy to prove by the Lemma 4.3. We omit further details. □

Claim 5.5. The distribution $\mathcal{D}_7$ is statistically indistinguishable from the distribution $\mathcal{D}_{real}$, where $\mathbf{U}' \leftarrow \mathbb{Z}_q^{t\times N}$ and $\mathbf{U} \leftarrow \mathbb{Z}_q^{n\times N}$, then we have that $\mathcal{D}_{real}$ is as follows:

$$\Big((\mathbf{U}'), \underline{\mathbf{U}}, \mathbf{M}_{(i,j)}\mathbf{SG}, \mathbf{S}'\mathbf{A} + \mathbf{E}, \mathbf{A} \mid (\mathbf{U}'), \underline{\mathbf{U}}, \sum(\mathbf{U}'), \sum(\underline{\mathbf{U}}), \sum((\mathbf{M}_{(i,j)}\mathbf{S})\cdot\mathbf{G}) \mid h(\mathbf{A}, t^*, e^*)\Big).$$

Obviously, from these claims which were described as above, we have that $\mathcal{D}_{origin} \approx_c \mathcal{D}_{real}$.

6 Conclusion

In this paper, we have attempted to give a practical solution to achieve multi-bit FHE scheme and prevent unauthorized users to tamper the data in the public cloud server by constructing a leakage resilient FHE on multi-bit message. Namely that, our multi-bit FHE is able to prevent the attacker from capturing the valid information of cryptography secret key via side channel attacks. For better comparison, we first briefly summarize the concrete key sizes of LWE-based multi-bit FHE schemes in Table 2. Our MGSW scheme achieves flexibility decryption, and it can decrypt at one-time. Particularly, Table 1 shows that, our MGSW scheme achieves leakage-resilience (i.e., LRMGSW scheme) and tolerates more leaked bits as compared with the leakage resilient HAO [9] scheme.

Moreover, our work leaves many interesting open questions that may be addressed in the future. For instance, our ideas may be useful to enhance the security of big data and construct oblivious transfer for one-time multi-bit exchange, which might be very important yet challenge.

Acknowledgments

The authors are grateful to the anonymous reviewers for their valuable comments that highly improve the readability and completeness of the paper. This work was supported by the National Natural Science Foundation of China (No.61472097).

References

[1] R. V. Prakash, "Big data analysis: Big data analysis pipeline and its technical challenges," *Effective Big Data Management and Opportunities for Implementation*. IGI Global, pp. 83–93, 2016.
[2] H. Wang, D. He, and S. Tang, "Identity-based proxy-oriented data uploading and remote data integrity checking in public cloud," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 6, pp. 1165–1176, Jun. 2016.
[3] D. Wang and P. Wang, "Two birds with one stone: Two-factor authentication with security beyond conventional bound," *IEEE Trans. Depend. Secur. Comput.*, 2016, Doi: 10.1109/TDSC.2016.2605087.
[4] C. Gentry, "Fully homomorphic encryption using ideal lattice," in *Proc. 41st Annu. ACM Symp. Theory Comput.*, 2009, pp. 169–178.
[5] Z. Brakerski, "Fully homomorphic encryption without modulus switching from classical gapSVP," in *Proc. 32nd Annu. Cryptology Conf. Advances Cryptology*, 2012, pp. 868–886.
[6] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based," in *Proc. Annu. Cryptology Conf. Advances Cryptology*, 2013, pp. 75–92.
[7] J. Alperin-Sheriff and C. Peikert, "Faster bootstrapping with polynomial error," in *Proc. Annu. Cryptology Conf. Advances Cryptology*, 2014, pp. 297–314.
[8] A. Berkoff and F.-H. Liu, "Leakage resilient fully homomorphic encryption," in *Proc. Theory Cryptography Conf.*, 2014, pp. 515–539.
[9] R. Hiromasa, M. Abe, and T. Okamoto, "Packing messages and optimizing bootstrapping in GSW-FHE," in *Proc. IACR Int. Workshop Public Key Cryptography*, 2015, pp. 699–715.
[10] P. Mukherjee and D. Wichs, "Two round multiparty computation via multi-key FHE," in *Proc. 35th Annu. Int. Conf. Advances Cryptology*, 2016, pp. 735–763.
[11] Z. Li, C. Ma, E. Morais, and G. Du, "Multi-bit leveled homomorphic encryption via dual.LWE-based," in *Proc. Int. Conf. Inf. Secur. Cryptology*, 2016, pp. 221–242.
[12] A. López-Alt, E. Tromer, and V. Vaikuntanathan, "On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption," in *Proc. 44th Annu. ACM Symp. Theory Comput.*, 2012, pp. 1219–1234.

[13] M. Clear and C. McGoldrick, "Multi-identity and multi-key leveled FHE from learning with errors," in *Proc. Annu. Cryptology Conf. Advances Cryptology*, 2015, pp. 630–656.
[14] Z. Brakerski and R. Perlman, "Lattice-based fully dynamic multi-key FHE with short ciphertexts," in *Proc. Annu. Cryptology Conf. Advances Cryptology*, 2016, pp. 190–213.
[15] C. Peikert and S. Shiehian, "Multi-key FHE from LWE, revisited," in *Proc. Theory Cryptography Conf.*, 2016, pp. 217–238.
[16] Z. Brakerski, C. Gentry, and S. Halevi, "Packed ciphertexts in LWE-based homomorphic encryption," in *Proc. IACR Int. Workshop Public Key Cryptography*, 2013, pp. 1–13.
[17] S. Micali and L. Reyzin, "Physically observable cryptography," in *Proc. Theory Cryptography Conf.*, 2004, pp. 278–296.
[18] A. Akavia, S. Goldwasser, and V. Vaikuntanathan, "Simultaneous hardcore bits and cryptography against memory attacks," in *Proc. Theory Cryptography Conf.*, 2009, pp. 474–495.
[19] B. Applebaum, D. Cash, C. Peikert, and A. Sahai, "Fast cryptographic primitives and circular-secure encryption based on hard learning problems," in *Proc. Annu. Cryptology Conf. Advances Cryptology*, 2009, pp. 595–618.
[20] S. Goldwasser, Y. T. Kalai, C. Peikert, and V. Vaikuntanathan, "Robustness of the learning with errors assumption," in *Proc. Conf. Innovations Theoretical Comput. Sci.*, 2010, pp. 230–240.
[21] C. Peikert, V. Vaikuntanathan, and B. Waters, "A framework for efficient and composable oblivious transfer," in *Proc. Annu. Cryptology Conf. Advances Cryptology*, 2008, pp. 554–571.
[22] R. Lindner and C. Peikert, "Better key sizes (and attacks) for LWE-based encryption," in *Proc. 11th Int. Conf. Topics Cryptology*, 2011, pp. 319–339.
[23] R. Impagliazzo, L. A. Levin, and M. Luby, "Pseudo-random generation from one-way functions," in *Proc. Annu. ACM Symp. Theory Comput.*, 1989, pp. 12–24.
[24] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," in *Proc. IEEE 52nd Annu. Symp. Found. Comput. Sci.*, 2011, pp. 97–106.
[25] V. Lyubashevsky, "Lattice signatures without trapdoors," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Techn.*, 2012, pp. 738–755.
[26] Y. Yu and J. Zhang, "Cryptography with auxiliary input and trapdoor from constant-noise LPN," in *Proc. Annu. Cryptology Conf. Advances Cryptology*, 2016, pp. 214–243.
[27] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM J. Comput.*, vol. 38, no. 1, pp. 97–139, 2008.
[28] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," in *Proc. Annu. ACM Symp. Theory Comput.*, 2005, pp. 84–93.
[29] C. Peikert, "Public-key cryptosystems from the worst-case shortest vector problem," in *Proc. Annu. ACM Symp. Theory Comput.*, 2009, pp. 333–342.
[30] C. Peikert and B. Waters, "Lossy trapdoor functions and their applications," in *Proc. Annu. ACM Symp. Theory Comput.*, 2008, pp. 187–196.
[31] J. Katz, A. Thiruvengadam, and H.-S. Zhou, "Feasibility and infeasibility of adaptively secure fully homomorphic encryption," in *Proc. IACR Int. Workshop Public Key Cryptography*, 2013, pp. 14–31.
[32] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*. Boca Raton, FL, USA: CRC Press, 2014.
[33] Z. Brakerski and V. Vaikuntanathan, "Lattice-based FHE as secure as PKE," in *Proc. 5th Conf. Innovations Theoretical Comput. Sci.*, 2014, pp. 1–12.
[34] D. Micciancio and C. Peikert, "Trapdoors for lattices: Simpler, tighter, faster, smaller," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Techn.*, 2012, pp. 700–718.
[35] L. Ducas and D. Micciancio, "FHEW: Bootstrapping homomorphic encryption in less than a second," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Techn.*, 2015, pp. 617–640.
[36] Z. Li, S. D. Galbraith, and C. Ma, "Preventing adaptive key recovery attacks on the GSW leveled homomorphic encryption scheme," in *Proc. Int. Conf. Provable Secur.*, pp. 373–383.
[37] J. Alwen, Y. Dodis, M. Naor, G. Segev, S. Walfish, and D. Wichs, "Public-key encryption in the bounded-retrieval model," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Techn.*, 2010, pp. 113–134.

Zengpeng Li is working toward the PhD degree at Harbin Engineering University, China. His primary research interests include cryptography, protocol, and information security. In particular, his research focus is lattice-based cryptography. He is a student member of the IEEE, the CCF, the IEICE, and the CACR (Chinese Association for Cryptographic Research).

Chunguang Ma received the PhD degree in cryptography from the College of Computer Science and Technology, Beijing University of Posts and Telecommunications, China, in 2005. He is currently a professor in the College of Computer Science and Technology, Harbin Engineering University. His main research interests include cryptography and information security, in particular, cryptographic protocols.

Ding Wang received the PhD degree in information security from Peking University in 2017. He is currently supported by the Boya Postdoctoral Fellowship in Peking University, China. He has been involved in the community as a TPC member and a reviewer for more than 50 international conferences and journals. His works appear in prestigious conferences like ACM CCS, IEEE/IFIP DSN, ESORICS, and journals like the IEEE Transactions on Information Forensics and Security and the *IEEE Transactions on Dependable and Secure Computing*. Three of them are selected as "ESI highly cited papers". His research interests include cryptography and system security.